Komputer 7/8


\lipiec - sierpień 1989\


popularny miesięcznik informatyczny: \# (40-41) 89\cena 230 zł.



Relacyjne bazy danych

Trzy wymiary



Nowe wirusy...

Plotery






Kurier

Komputer w domu

Komputer w pracy

Mikromarket

Szef kuchni poleca:



Po raz pierwszy zasiadając do pisania comiesięcznego felietonu otwierającego nasz miesięcznik czuję się nie w porządku w stosunku do Czytelników. Pierwszy raz bowiem dostaniecie Państwo numer o normalnej objętości, za to o podwójnej numeracji i datowany lipiec/ sierpień. I nie sądzę, aby satysfakcjonowało Państwa wyjaśnienie, iż decyzja ta została redakcji narzucona przez wydawcę, którego z kolei zmuszają do tego tak zwane trudności obiektywne. Nie pozostaje mi nic innego, jak tylko przeprosić za, delikatnie mówiąc, zmianę zasad gry w trakcie meczu (ostatnio taka praktyka stała się w naszym kraju niemal regułą).

Numer wakacyjny staraliśmy się uczynić nieco lżejszym, co wszakże nie oznacza gorszym – jest w nim więcej niż zazwyczaj opisów nowych gier i drobnych programów. Na początek proponuję jednak rozmowę z przedstawicielem firmy Microsoft. Tytuł wywiadu ***Stawiamy na ewolucję***.

Uczestników i sympatyków konkursu ***Mikro Historicus*** zainteresuje zapewne prezentowany w tym numerze pierwszy spośród nagrodzonych programów.

O mikrokomputerze Elwro 800 Junior napisano wiele złego, także na tych łamach, tym niemniej znajduje się on w niejednej szkole i należałoby coś z nim zrobić. Myślę, że przed nowym rokiem cenne będą konkretne i praktyczne uwagi pana Rudolfa, nauczyciela z Otmuchowa (***Z taśmy na dyskietkę***).

Wirusy atakują twierdzi Andrzej Kadlof i nikogo dziś już nie trzeba przekonywać o niebezpieczeństwie. Więcej, dające się zauważyć objawy paniki i nerwowe działania użytkowników powodują dodatkowe szkody wynikające z podejmowania nie do końca przemyślanych środków zaradczych. Autor artykułu przedstawia kilka spośród spotykanych w kraju wirusów (w następnych numerach temat będzie kontynuowany).

Relacyjne bazy danych, to artykuł dla Czytelników o pewnym przygotowaniu matematycznym i informatycznym, a więc raczej trudny, niemniej ze względu na ważność tematu wart przeczytania.

W tym numerze po raz pierwszy pojawiła się winieta ***Klinika oprogramowania***. Otworzyliśmy w ten sposób nową, stałą rubrykę, do lektury której gorąco zachęcam. Poświęcamy ją dostosowywaniu do potrzeb polskiego użytkownika gotowych, istniejących już programów. Doświadczenie wskazuje bowiem, że niewielkie niekiedy zmiany w programach mogą znacznie podnieść ich walory użytkowe.

Na koniec zwracam uwagę na artykuł Zbigniewa Blewońskiego zatytułowany ***Plotery***. Ci, którzy zetkną się z tym tematem po raz pierwszy otrzymają porcję podstawowych wiadomości, dla pozostałych będzie to okazja do usystematyzowania i zweryfikowania wiadomości.

Życzę Państwu przyjemnej lektury i udanych wakacji.

Stanisław Marek Królak

7/8 (40)

"Komputer" Popularny Miesięcznik Informatyczny - pismo miłośników i użytkowników mikrokomputerów redagują:

Marek Młynarski (red. nacz. tel. w. 330)
Grzegorz Eider (z-ca red. nacz. tel. w. 330)
Stanisław M. Królak (sekr. red. tel. w. 330)
Irena Urbaniak (z-ca sekr. red. tel. w. 330)
Kierownicy działów:
Marek Car (publicystyka tel. w. 329)
Małgorzata Luzińska (dział tech. tel. w. 310)
Krzysztof Matey (komputery domowe tel. w. 329)
Zenon Rudak (sprzęt tel. w. 310)
Stefan Szczypka (dział graf. tel. w. 329)
Tomasz Zieliński (programy tel. w. 310)
Dziennikarze:
Grzegorz Czapkiewicz
Mariusz Dec
Piotr Kakiet
oraz zespół:
Zbigniew Blewoński, Andrzej Kadlof, Tomasz Mazur, Wiesław Migut, Juliusz Rawicz, Leszek Rudak, Michał Setlak, Jakub Tatarkiewicz, Roland Wacławek (Katowice) i współpracownicy: Maciej Borkowski (Poznań), Tadeusz Jedynak (Tarnowskie Góry), Jarosław Kania, Zbigniew Kasprzycki, Marek Matuszczak, Mariusz Pietruszka (Tarnowskie Góry), Jan Stożek (sysop), Tadeusz Wilczek, Andrzej Załuski (Kraków)

Magdalena Stachorzyńska (operatorka komputera)
Korekta: Maria Omiecińska,

Adres redakcji
ul. Koszykowa 6A,
00-564 Warszawa,
Telefony
21-19-85 lub
centrala 28-22-01
wew. 243 lub 328
telex 812405 ruch pl

Sieć FIDO
211985 w godz. 16^{00} – 10^{00}
soboty i niedziele całą dobę.
Wydawca: Warszawskie Wydawnictwo Prasowe RSW „Prasa-Książka-Ruch", Al. Jerozolimskie 125/127,
dyr. Maciej Hoffman
02-017 Warszawa, tel. centrali: 28-52-31.
Skład i druk: Prasowe Zakłady Graficzne, Łódź, ul. Armii Czerwonej 28.
Cena: 230 zł. Zam. 1683/89, A-63.

Nr indeksu 36-345 ISSN 0860-2514
Dyskietkę z tekstami do numeru przekazano do składu w dniu 10.05.1989.

\Kurier\

Marek Młynarski

Zabiegi magiczne

Idź o północy na rozstaje dróg, pod stary dąb, w który trafił piorun i czekaj na pohukiwanie puszczyka. Musi być wtedy nów księżyca, a gdy usłyszysz ptaka, szybko wysyp za lewe ramię startą na proszek suszoną żabę wymieszaną ze śliną nietoperza. Weź teraz dwa numery "Komputera" i trzyj je wzajemnie tak długo, aż będą stanowiły, nieokreśloną masę. Wówczas wypowiedz magiczne zaklęcie i pozostawiając na rozstajach swoje plany szybko odejdź, nie oglądając się za siebie. Po kilku miesiącach otrzymasz to, czego w żadnym wypadku nie chciałeś, ale przed czym uchronić się nie można, bowiem nie masz żadnych szans wpłynięcia na to, co robią inni.

Sądząc po efektach, właśnie taki zabieg magiczny wykonała cała nasza redakcja, nie bacząc na lejący deszcz i piski przerażonych dziewcząt. Muszę jednak rozczarować Czytelników – połączenie numerów 7 i 8 obyło się bez opisanej scenerii, chociaż efekt jest taki, jak większości zabiegów magicznych – dokładnie żaden. Nie przyspieszy to wydawania pisma i chociaż w tym roku kolejne numery będą może bardziej zgodne z numerami miesięcy, w których powinny się ukazywać, nie zmieni to faktu, że wydań "Komputera" w 1989 roku będzie już tylko jedenaście.

Przepraszam za to Czytelników, i jedyne, co mogę w imieniu całej redakcji powiedzieć, to powtórnie stwierdzić, że zrobiliśmy wszystko dla utrzymania terminowości wydawania pisma. W fatalnych czasach, w których przyszło nam działać czyste sumienie nie jest walutą ani nawet złotówką najmocniejszą, ale takie właśnie poczucie towarzyszy naszym redakcyjnym poczynaniom.

Nie jest to niestety koniec zabiegów magicznych, o których pragnę dzisiaj poinformować. Już wiadomo o następnym, wywołanym przez powszechną i totalitarną podwyżkę cen na wszystkie związane z drukiem pisma materiały i usługi. Ceny papieru, koszty druku, wydatki na transport, to, co trzeba zapłacić za sprzedanie pisma w tysiącach kiosków – wszystko to jest dziś niepomiernie droższe niż jeszcze wczoraj. Już to kiedyś pisałem, ale powtórzę jeszcze raz, najtańszym "elementem" tych wszystkich kosztów są płace zatrudnionych w redakcji ludzi.

Efekt wzrostu kosztów jest łatwy do przewidzenia – wzrośnie cena "Komputera" (na razie do 350 złotych) i to już od przyszłego numeru. Proponuję jednak porównanie, jak ten wydatek, ponoszony raz na miesiąc ma się np. do ceny pudełka papierosów. Zabieg podnoszenia ceny ma oczywiście bardziej ukrytą część magiczną niż zabieg połączenia numerów. Wyższa cena pozwala na egzystencję pisma, ale w skali całej gospodarki jest to też zabieg magiczny – nic właściwie nie zmieniający poza tłamszeniem w kieszeniach banknotów o coraz wyższych nominałach, przy narastającej praktycznej pustce tychże kieszeni.

Kolejny zabieg magiczny, wart opisania, związany jest już ściśle z komputerami, a dotyczy ochrony praw autorskich. Rozmaite wielce szacowne gremia podjęły w tej sprawie stanowczo brzmiące uchwały, stanowiska czy też po prostu Głosy w Dyskusji. My zresztą też nie odmówiliśmy sobie tej przyjemności i wcale nie mamy zamiaru nie zabierać dalej głosu. Ba, przy Wydawnictwach Naukowo Tehchnicznych utworzono Radę, której zadaniem ma być ochrona oprogramowania. Problemem tym zainteresowany był także ZAIKS, z którego przedstawicielem odbyłem długą rozmowę. Co ciekawe, wszyscy są zgodni, że tak dalej to już naprawdę być nie może.

Niestety jest to znów typowy zabieg magiczny, tym razem z gatunku "głos wołającego na puszczy". Nic bowiem z owych uchwał, rezolucji i głosów konkretnego nie wynika, bo i wyniknąć nie może. Po pierwsze – każdy sobie rzepkę skrobie. Po drugie – organy teoretycznie powołane do rozwoju i ochrony naszej myśli, w tym i technicznej, po smacznym śnie obwarowały się w swych Urzędach ze strachu przed zmianami. Po trzecie i chyba wreszcie najważniejsze – Polska tak daleko odjechała od reszty świata, że nawiązanie rozmowy jest bardzo trudne, a kto wie czy w ogóle możliwe bez tłumacza. A kto może być tłumaczem i co z tym fantem dalej?



\Kurier\

Grzegorz Eider

Bez scenariusza

Polski rynek mikrokomputerowy bywa przyrównywany do rynku owocowo-warzywnego, który w naszym kraju jest symbolem wszystkiego, co pozytywne w gospodarce rynkowej. Przyrównanie do pietruchy stanowi w Polsce nobilitację, nawet dla komputera (czyż to nie zabawne?). Rynek komputerowy w pełni na tę nobilitację zasługuje. Ceny – jeśli uwzględni się inflację – od lat maleją, kupujący może przebierać i grymasić – w kolejce zawsze czeka gromada dostawców. Raj?

Wszystkie te pozytywne cechy rynku mogły zaistnieć dzięki bardzo silnej konkurencji. Można powiedzieć, że polski rynek komputerowy jest jakby żywcem wyjęty z podręcznika ekonomii, z rozdziału: "Wczesny kapitalizm" – całkowicie rynkowe zasady, rozdrobnienie, brak monopoli, mobilny kapitał (oczywiście na skalę odpowiadającą istniejącemu rozdrobnieniu), to atrybuty idealnego modelu gospodarki rynkowej, ze wszystkimi jej zaletami, które z taką lubością obserwujemy. Jednak naszkicowany obraz jest sielanką tylko z pozoru – zakłócają go dwa niezwykle ważne czynniki.

Ludowe przysłowie powiada, że wystarczy łyżka dziegciu, by zmarnować beczkę miodu. Nikt jakoś nie zadaje sobie pytania czy ta beczka miodu wystarczy, by zniszczyć łyżkę dziegciu? Czy można kroplę – rynek komputerowy wlać do oceanu gospodarczych nonsensów bez szkody dla kropli? Pierwszym elementem dewastującym sielankę jest konieczność współistnienia rynku mikrokomputerowego z bynajmniej nie rynkowym (w każdym razie nie w rozumieniu klasycznego, idealnego modelu) otoczeniem gospodarczym. Nacisk tego otoczenia stanowi stale zagrożenie dla rynku komputerowego. Wprawdzie wydaje się, że czasy administracyjnych zakazów- nakazów w skali mogącej zlikwidować całą branżę – mamy już chyba za sobą. Wprawdzie wydaje się, że dryfujemy (lub żeglujemy – to zależy od interpretacji) w stronę gospodarki rynkowej. Wprawdzie w którą byśmy stronę nie skierowali głowy – zewsząd słychać tokowanie reformatorów. Otóż mimo tych wszystkich "wprawdzie" na co dzień mamy do czynienia z gospodarczym dziedzictwem minionych "epok" – rozklekotaną maszyną gospodarki. To nie są puste słowa czy też zgrabne, ale pozbawione treści porównania. Exemplum: w Polsce nie istnieje de facto system bankowy (rozumiany czysto funkcjonalnie – jako sieć banków zapewniających sprawną obsługę finansową gospodarki i obywateli). Parę banków wprawdzie jest, ale po pierwsze zbyt mało... (dalsze wyliczania nie są konieczne skoro nie ma armat). To element naszej zgrzytającej maszynerii. Na rynku pojawia się duża liczba nowych firm (w tym komputerowych) a tym samym transakcji. Banki nie są w stanie temu sprostać. Powstają zaległości. Są firmy mające spory zysk "w papierach" i debet na koncie (choć czasami dłużnicy już dawno zaległości zapłacili – tyle że przelewy tygodniami "wędrują" z banku do banku, z oddziału do oddziału, z wydziału do wydziału...). W warunkach ogromnej inflacji (też dziedzictwo) stracone dni to wymiernie stracone pieniądze. Nagminna więc staje się praktyka nie płacenia i wmawiania wierzycielowi, że się zapłaciło, tylko "pewnie w banku jeszcze nie zaksięgowali...". W rezultacie istniejące warunki promują (selekcja naturalna) postawę typową dla hosztaplera, a nie rzetelnego kupca. Gdyby banki działały sprawnie, efekt selekcji nie występowałby (a w każdym razie byłby o wiele słabszy).

Konfrontacja zdrowego (jak się wydaje przynajmniej na pierwszy rzut oka) rynku mikro z chorym otoczeniem trwa. W jej efekcie rynek komputerowy okazuje się nie taki już wzorcowy.

Jest i element drugi kalający nam sielankowy obraz. Otóż klasyczny model idealnego rynku, taki wczesny kapitalizm, jeśli w ogóle występował, to w innych czasach – przede wszystkim technologicznie. Przykład pierwszy z brzegu. Każdy kupiec – w naszym idealnym modelu – zachwala swoje towary, dajmy na to młotki. Przychodzi klient, wysłuchuje zachwalającej mowy kupca, bierze młotek do ręki i... po pierwsze wie co z nim zrobić, po drugie potrafi go ocenić. Teraz przejdźmy do straganu z komputerami. Tu kupujący np. PC/AT nie jest w stanie ocenić nie tylko jakości produktu, ale także – na ogół – nie potrafi wyartykułować swoich potrzeb. Jest bezradny i zdany na kompetencję oraz rzetelność kupca. Tymczasem kupiec zachowuje się zgodnie z modelem – zachwala towar. Wiele firm dostarczających sprzęt twierdzi (pisałem o tym przy okazji omawiania najnowszej edycji Katalogu Firm Komputerowych), że prowadzi doradztwo czyli consulting. Jest to oczywisty absurd a właściwie próba (nieświadoma?) oszukania klienta: niech pan przyjdzie, doradzimy panu nie tylko co kupić, ale także jakie są najlepsze oferty.

Tak, ten sposób funkcjonowania rynku – rodem z bazaru, właśnie wspomnianego na wstępie "zielonego rynku" – nie jest adekwatny do hi-tech będącej jego przedmiotem. A jednak taki model funkcjonuje, chcemy tego czy nie. Przyczyna jest prosta – branża rodziła się bez kapitału i funkcjonuje w warunkach nie skłaniających do jego akumulacji. Status quo samo się więc podtrzymuje. Wyjściem byłaby koncentracja kapitału (może pojawienia się na rynku kapitału zagranicznego?) oraz wyraźna specjalizacja (consulting powinny prowadzić wyspecjalizowane firmy, niezależne od dostawców sprzętu i oprogramowania).

Podsumujmy tę, nieco przydługą tyradę. Teza, że rynek komputerowy, podobniej jak rynek owocowo-warzywny, jest calkowicie normalnym rynkiem (czytaj: normalnym w sensie modelu idealnej gospodarki rynkowej) jest fałszywa. Przede wszystkim nie ma modelowego, idealnego rynku komputerowego bowiem branża ta tworzyła się i rozwijała w rodzącym patologiczne dewiacje otoczeniu. I to jest pierwsza skaza na potocznym wyobrażeniu o światku mikro. Nawet gdybyśmy jednak mieli do czynienia z modelem idealnym, to okazuje się, że nie przystaje on do wymogów, jakie tworzy sam przedmiot funkcjonowania tego rynku – komputery. Rozwój branży wymaga koncentracji kapitału (również koncentracji organizacyjnej), wymaga zdecydowanej specjalizacji, wymaga nowoczesnego zarządzania i marketingu.

Przyznam, ze jestem pesymistą. Scenariuszy, które potrafię sobie wyobrazić jest kilka. Żaden jednak nie wygląda optymistycznie.



Na 10 dni przed drukiem

Kolejny wirus

Inwazja trwa. W pierwszych dniach lipca pojawił się w Warszawie nowy okaz o roboczej nazwie Ping-Pong. Niepotwierdzone doniesienia sugerują, że dręczy już także użytkowników w innych rejonach Polski. Widocznym objawem infekcji jest pojawiająca się od czasu do czasu na ekranie "latająca piłeczka" oraz błędne wczytywanie niektórych plików (w tym również programów). Jest to pierwszy schwytany wirus, który nie atakuje plików, a mimo to przenosi się z dyskietki na dyskietkę lub twardy dysk. W czasie infekcji zmienia zawartość sektora inicjującego oraz znakuje jedno grono (cluster) jako błędne i umieszcza w nim resztę swego kodu. W czasie startu systemu operacyjnego instaluje część rezydentną i infekuje każdą dyskietkę z której cokolwiek się czyta. Prawdopodobnie nie niszczy żadnych zasobów użytkownika, ale współżycie z nim może być kłopotliwe. Redakcja dysponuje szczepionką przeciwko temu wirusowi (DOCTOR.EXE). *[Andrzej Kadlof]*

Prix Ars Electronica'89

Jury dorocznego Międzynarodowego Konkursu Sztuki Komputerowej opublikowało listę nagrodzonych prac. W konkursie sponsorowanym przez Austriackie Radio i Telewizję oraz koncern Siemens AG wzięło udział 551 artystów z 29 krajów, nadesłano przeszło 1100 prac. Dzięki uprzejmości współtwórcy koncepcji i organizacji konkursu – Dr Christine Schopf mamy możność poinformowania, że w kategorii animacji komputerowej Złotą Nike otrzymała Joan Staveley z USA za pracę "Broken Heart". Złota Nike w kategorii grafiki komputerowej przypadła za pracę "Gramophone" Węgrowi, Tomas'owi Walicky'emu. W dziedzinie muzyki Złota Nike powędrowała do Finlandii – utwory "Io" i "Stilleben" Kaija Saariaho. Nagrody rozdano 13 września w Linzu w Austrii.



Twórcy sztuki komputerowej zainteresowani konkursem otrzymają informacje o tegorocznej i przyszłej edycji u p. Zdzisława Pokutyckiego – Studio Artystyczne "VISIO", 53-638 Wrocław 57-92, tel. (0-71) 551699 *[grei]*

MIKROLAUR '90

Przypominamy, że tylko do 30 listopada można zgłaszać prace na konkurs MIKROLAUR '90. Bliższe informacje i regulamin zamieściliśmy w numerze 4/89 str. 18.

Nagroda ta, przyznawana za najwybitniejsze osiągnięcia w dziedzinie oprogramowania i sprzętu komputerowego, jest najbardziej prestiżowym wyróżnieniem na polskim rynku komputerowym. Prace laureatów prezentowane są w naszym miesięczniku. Gorąco nawiamy do udziału ! *[T.Z.]*

Oj Amstradzie, Amstradzie ...

Najpierw opóźnienia dostaw tłumaczono brakiem układów DRAM (co zresztą było prawdą), potem (z początkiem lata) okazało się, że firma Amstrad po dwu latach musi przestrzegać ograniczeń narzuconych przez COCOM, zabraniających m.in. eksportu do krajów Układu W-skiego komputerów zawierających w sobie więcej niż 10% części amerykańskich. Wreszcie okazuje się, że notowania akcji Amstrada spadły na londyńskiej giełdzie z 230p do 74p. Jest to giełdowy obraz głębokiego kryzysu, jaki obecnie przechodzi firma Amstrad. Dotychczasowy przedstawiciel Amstrada na Polskę, A. Łukomski sądzi, że oznaczać to może kres interesów u nas. Czy zniesienie części restrykcji COCOM w lipcu br. wpłynie na poprawę sytuacji finansowej Amstrada ? Przyczyny kryzysu leżą tym razem jednak na Dalekim Wschodzie, a nie Wschodzie Najbliższym.

Sprostowania, przeprosiny, informacje

Przez ten tak długi czas, kiedy to drukarnia nijak nie mogła zmobilizować się do druku "Komputera", nazbierało się sporo drobnych, ale koniecznych informacji:

♦ S.Jaskólski, Dyrektor ROEM Ltd. oddział w Polsce informuje, że nie jest prawdą stwierdzenie, iż wymieniona w wywiadzie przeprowadzonym przez W. Majewskiego z dyr. handlowym Quest Automation Ltd. jest "wyłącznym dystrybutorem wyrobów firmy Noveil na rynku polskim". Byłoby to sprzeczne z generalną zasadą przeciwdziałania monopolizacji rynku. Firma ROEM Ltd. od 1988 r. ma uprawnienia do sprzedaży wyrobów Novell jako atoryzowany dystrybutor.

♦ Informowaliśmy o inicjatywie wydawania biuletynu dyskowego mikroPRESS. Nie napisaliśmy niestety nic o zespole redagującym. Przepraszamy i informujemy, że biuletyn redagują: P. Niezgodzki (red. naczelny), J. Rudowski, K. Gajewski, E. Mańkiewicz-Cudny, J. Niezgodzka. *[grei]*

♦ Andrzej Popławski, autor atrykułu "Polskie znaki raz jeszcze" ("Komputer" 6/89) przeprasza za błąd i prosi o wprowadzenie poprawek w wydruku "POLNQ STAR NL 10". We wszystkich przypadkach występującą na początku każdej linii liczbę 23 należy zastąpić liczbą 27 (łącznie 16 razy).

♦ Pana Jacka Jedynaka, który przetłumaczył nagraną na kasetę magnetofonową relację z naszej wizyty w fabryce ESSEX (ESSEX w Singapurze, "Komputer" 5/89) przepraszamy za pominięcie jego nazwiska w publikacji.

♦ Znowu drożej ! Warszawskie Wydawnictwo Prasowe informuje, że w związku ze wzrostem kosztów zmuszone jest podnieść cenę "Komputera" do 350 zł, od numeru 9/89. Nie będzie to niestety kres wzrostu ceny miesięcznika w br. Prenumeratorzy jak zwykle są wygrani, otrzymują pismo po dotychczasowej cenie do końca opłaconej prenumeraty.

Kolumnę przygotował 22 września 1989 r. Marek Młynarski.



Z przedstawicielem firmy Microsoft Reinerem Michlem rozmawia Tomasz Zieliński.

– Jaki jest cel pańskiej wizyty w Polsce?

– Wiąże się ona przede wszystkim z inauguracją działalności pierwszego oficjalnego dystrybutora naszego oprogramowania w Polsce – ZETO Wrocław. Ponadto staram się o podtrzymanie dwuletnich już kontaktów z różnymi firmami w Polsce.

– Z jaką polską firmą nawiązywał pan pierwsze kontakty?

– Pierwszym poważnym klientem w Polsce było Elwro, z którym w 1987 roku podpisaliśmy umowę licencyjną na system operacyjny MS-DOS 3.3.

– Teraz zamierzacie rozpocząć dystrybucję w Polsce programów aplikacyjnych.

– Osiągnęliśmy z ZETO Wrocław porozumienie w sprawie sprzedaży naszego oprogramowania na rynku polskim – również za dolary. Na razie w wersji angielskiej. Całe przedsięwzięcie właśnie rusza. Chcemy, aby nasze programy były również w Polsce sprzedawane z pełną obsługą posprzedażną. Chodzi o to, by klient mógł bez problemów otrzymać u naszego polskiego dystrybutora np. najnowszą wersję kompilatora C.

W przyszłości chcemy nie tylko sprzedawać oprogramowanie, ale także szkolić użytkowników – przypuszczalnie wespół z innymi polskimi firmami.

– Wielu użytkowników mikrokomputerów profesjonalnych chętnie użyłoby pełnej, najnowszej wersji systemu operacyjnego. Czy przewiduje pan sprzedaż MS-DOS-u także indywidualnym użytkownikom?

– Tę sprawę już dyskutowaliśmy z ZETO. Jeżeli zainteresowanych będzie więcej, to postaramy się znaleźć jakieś rozwiązanie. Na razie licencjonowany system operacyjny Microsofta można kupić w Polsce tylko ze sprzętem z Elwro.

– Czy przewiduje się przygotowanie polskich wersji waszych programów?

– Tak, mamy takie zamiary. Jest to zresztą wszędzie jeden z kluczy do sukcesu. Takie produkty jak

Microsoft

MS-DOS, WORKS, EXCEL czy WINDOWS i oczywiście dokumentacja – także do najnowszych kompilatorów – będą przetłumaczone na język polski. Jako pierwszą – najprawdopodobniej – zaoferujemy w drugiej połowie bieżącego roku polską wersję MS-DOS-u.

W przyszłości naszym celem będzie utworzenie filii w Polsce. Będzie to jednak uzależnione od wielkości obrotów.

– Jak firma zamierza zagwarantować sobie w Polsce ochronę sprzedawanych programów?

– Uzgodniliśmy z naszym dystrybutorem, że każdy nabywca będzie podpisywał specjalną umowę, w której zobowiąże się i zagwarantuje, że nie będzie kopiował zakupionego programu. Myślę, że będzie to skuteczne.

– Z pewnością orientuje się pan, jak wygląda u nas ochrona prawna przed nieautoryzowanym kopiowaniem oprogramowania? Czy i jak zamierza pan walczyć z piractwem?

– Myślę, że nikt w Polsce nie jest zadowolony z istniejącej sytuacji. Nie chodzi nawet o samego Microsofta, który jest relatywnie bogatą firmą amerykańską i nie czerpie wszak głównych zysków z handlu z Polską. Jest to niekorzystne przede wszystkim dla polskich informatyków, którzy chcieliby sprzedawać swoje programy. Wierzę, że ta sprawa zostanie w Polsce prawnie uregulowana. Tym bardziej, że zauważalny jest pewien postęp w świadomości ludzi, którzy coraz częściej zwracają uwagę na to, czy sprzedaje się im legalne oprogramowanie.

Chcemy w tym pomóc dając użytkownikom sprzętu komputerowego możliwość legalnego zakupu naszego oprogramowania. Oczywiście po gwarantowanej cenie, jak wszędzie na świecie. Nie powinno być przypadków zawyżania cen, co w przeszłości niejednokrotnie miało miejsce.

Równocześnie musimy rozmawiać z tymi, którzy kopiują nasze oprogramowanie i sprzedają je nielegalnie, starając się ich przekonać, czyli rozwiązać ten problem "pokojowo". Gdyby jednak okazało się to niemożliwe, są inne środki. O odwołaniu się do nich nie myślimy na razie poważnie.

– Firma ma kontakty w innych krajach socjalistycznych. Czy tam też są kłopoty z przestrzeganiem praw autorskich?

– Mamy dość szerokie kontakty na Węgrzech, jedynym kraju Europy Wschodniej, gdzie prawa autorskie są przestrzegane. I chyba nie bez przyczyny, bo szereg węgierskich firm eksportuje swoje oprogramowanie, także na Zachód. W Jugosławii sytuacja jest daleka od zadowalającej. W ZSRR czynione są pewne kroki. Myślę, że niebawem ukaże się tam odpowiednia ustawa. Podobnie jest w Czechosłowacji. W NRD sprawa jest bardziej skomplikowana, gdyż na skutek silnego scentralizowania gospodarki ochrona praw autorskich oprogramowania jest tam rozstrzygnięciem natury politycznej. W Bułgarii odpowiednie prawo zostanie najprawdopodobniej wydane po tym, gdy zapadną regulacje w Związku Radzieckim. Jaki jeszcze kraj pominąłem?

– Rumunię.

– O Rumunii nie mogę nic powiedzieć. Nie prowadzimy żadnej działalności w tym kraju.

– Jak zatem prezentujemy się na tle innych krajów socjalistycznych?

– Trzeba odróżnić dwie sprawy. Jedna – to świadomość nieetyczności nielegalnego kopiowania programów. Druga – to konkretne kroki zmierzające do objęcia oprogramowania ochroną prawną. Myślę, że coraz więcej Polaków zdaje sobie sprawę z tego, że używa kradzionego oprogramowania. Natomiast przekonanie o potrzebie korzystania z oprogramowania legalnego zaczyna się dopiero kształtować. Również dzięki temu, że polskie programy znajdują rynek zbytu na Zachodzie.

– Jak pan szacuje stosunek kopii legalnych i nielegalnych waszych programów, np. systemu MS-DOS?

– W Polsce można chyba mówić o stosunku 5:95, tzn. 5% kopii legalnych, a 95% nielegalnych.

– A np. w RFN?

– W Niemczech Zachodnich naszych interesów broni bardzo dobrze funkcjonująca ustawa o ochronie oprogramowania. Tam stosunek kopii legalnych do nielegalnych określiłbym jak 80% do 20% czy 90% do 10%. Bardzo podobnie jest w USA i wielu innych krajach zachodnich.

– Czy wasza polityka w sprawie nielegalnego kopiowania programów jest tam tak samo elastyczna jak w Polsce?

– Tak, Microsoft jest firmą przyjacielską. Zresztą jest to jedna z naszych dewiz. Jeden z poprzednich naszych szefów mawiał: ***"Be friendly, be Microsoft like"***.

Jednakże jeżeli ktoś np. w RFN wytwarza nielegalne kopie w celu ich odsprzedaży, a my dowiemy się o tym procederze, to możemy go zaskarżyć do sądu. Były już przypadki, że policja włączała nas do prowadzonych śledztw i takie osoby znalazły się potem w więzieniu.

Natomiast w krajach, gdzie nie ma odpowiednich norm prawnych, staramy się perswadować. Jeżeli nie przynosi to rezultatów – uruchamiamy swoich prawników z ***Legal Department.***

– Ile firm na świecie zajmuje się sprzedażą waszego oprogramowania?

– Trudno mi powiedzieć. W większości krajów mamy przynajmniej jednego dystrybutora. Ale np. w RFN jest ich trzech, podobnie w Austrii i Szwajcarii. Myślę że w sumie będzie to kilkaset firm.

– A jak jest zorganizowana sprzedaż programów w innych krajach socjalistycznych? Czy firma ma tam swoich dystrybutorów?

– Mamy dystrybutorów na Węgrzech, w Bułgarii, w ZSRR, Jugosławii. W tych krajach jesteśmy w trakcie budowy sieci sprzedażnej naszych programów. W Czechosłowacji i NRD nie mamy jeszcze nikogo.

– Powiedział pan, że współpracę z Elwro Microsoft zapoczątkował przed dwoma laty. Ale wrocławski potentat nie jest jedynym producentem komputerów PC w Polsce.

– Myślę, że Elwro jest jedną z ważniejszych firm na polskim rynku. Cieszę się, że umowę w sprawie MS-DOS-u udało się nam podpisać właśnie z nimi. Było to ważne dla obu stron. Jest to jednak dopiero pierwszy krok. Będziemy przekonywać także innych producentów o potrzebie wyposażania sprzętu w licencjonowane kopie systemu operacyjnego.

– Czy Elwro było jedynym przekonywanym o sensowności takiego posunięcia?

– Prowadziliśmy rozmowy z wieloma innymi firmami. Były wśród nich takie, które na razie nie dały się przekonać do proponowanych przez nas rozwiązań. Nie będę ich wymieniać, nie byłoby to chyba dla nich zbyt korzystne. Wierzę jednak, że nawet one z czasem się zdecydują na legalne wyposażanie swoich komputerów w system MS-DOS.

Wiele polskich firm zamierza sprzedawać mikrokomputery w ZSRR. Nie będą mogły tego robić nie wyposażając sprzętu w legalny MS-DOS.

– Jest więc pan optymistą?

– Wynika to trochę z mentalności. Jestem odpowiedzialny za sprzedaż, czyli optymizm powinien być moją cechą zawodową. Jestem też optymistą dlatego, że przeprowadziłem tutaj wiele rozmów z ludźmi podzielającymi moją opinię, że rynek oprogramowania w Europie Wschodniej jest bardzo perspektywiczny. Microsoft będzie się starał mocno na nim uplasować. Mam nadzieję, że miesięcznik "Komputer" będzie mógł zaprezentować na swoich łamach oferowane przez nas oprogramowanie.

– Bardzo chętnie skorzystamy z okazji. Dziękuję za rozmowę.

Kurier

M. von der Vlaass

Rodzina "Kronos"

Zgodnie z zapowiedzią w poprzednim numerze naszego pisma - proponuję lekturę materiału, który dotarł do nas aż z Nowosybirska. Wielu Czytelników uzna jego przydatność za znikomą, być może. Pozostają jednak bezsporne walory poznawcze. Chcę przedstawić ludzi, którzy brak dostępu do komputera uznali za wystarczający powód, by go samemu zbudować.

Materiał dedykuję tym wszystkim "majsterklepkom", którzy nie tak dawno przecież z wypiekami na twarzy ślęczeli nad "Audio-Video" próbując, zgodnie z zamieszczanym tam opisem, zbudować własny komputer. Takich samych zapaleńców składających z części dostępnych na własnym rynku kopie Spectrum spotkałem w kwietniu br. w sercu Syberii. I wierzcie mi, maszynki te wyglądają o wiele lepiej niż pierwszy, własny komputer znanego Wam z naszych (i nie tylko) łamów Rolanda Wacławka. Przynajmniej obudowę mają z plastyku, a nie drewna. Ale to już zupełnie inna historia.

Marek Car

Kronos to ogólna nazwa rodziny 32-bitowych procesorów zorientowanych na języki wysokiego poziomu, przeznaczonych do wykorzystania w konstrukcjach komputerów o otwartej architekturze: od wbudowanych mikrokomputerów i jednoprocesorowych stanowisk typu workstation aż do wieloprocesorowych maszyn klasy super-mini.

Historia narodzin

W roku 1984, w pokoju domu akademickiego Uniwersytetu Nowosybirskiego urodził się Kronos. Dokładniejszej daty narodzin brak: trudno zdecydować, czy ma to być dzień, w którym spotkali się twórcy – studenci fizyki i matematyki – i powstał pomysł stworzenia maszyny z własnym systemem operacyjnym, czy też moment, gdy procesor został uruchomiony.

Natomiast z całą pewnością wiadomo, kto jest dziadkiem. To Niklaus Wirth, profesor z Zurichu, który stworzył język programowania Modula-2 i zrealizował go na swojej maszynie Lilith. Język Modula-2 przejął najlepsze cechy Pascala, ale ma szereg cech specyficznych, dzięki którym jest niezastąpiony przy tworzeniu oprogramowania. Przede wszystkim jest to modułowość (oraz możliwość podziału modułu na część definiującą i część wykonawczą), rozbudowane struktury danych i sterowania, precyzyjna kontrola typów, obecność procedur – co pozwala na dynamiczne wprowadzenie parametrów procedur z zewnątrz, oraz obecność środków programowania niskiego poziomu, pozwalających na osłabienie ostrej kontroli typów i odwzorowywanie danych strukturalnych w pamięci.

Z artykułem Wirtha zapoznał jednego z programistów jego kierownik naukowy, Igor Pottosin. I gdy Dima Kuzniecow (tak się nazywał ów programista) pisał swą pracę roczną – kompilator języka Modula-2 dla interpretera systemu rozkazów Lilith – zrodziła się u niego myśl o zbudowaniu własnej maszyny z systemem rozkazów, zorientowanym na ten wspaniały język. W tym właśnie czasie odbyło się wspomniane spotkanie z fizykami, którzy z kolei też chcieli zbudować komputer – i to nie jeden – do własnego użytku.

Również wtedy młodzi ludzie spotkali się z Aleksandrem Marczukiem, który później zaprosił ich do pracy w zespole "Start", ale o tym później.

Kapitał pierwotny – nie działająca "Elektronika-60" – zbierany był po kawałku w różnych instytutach Syberyjskiego Oddziału Akademii Nauk (SOAN) ZSRR, w których pracowali zainteresowani: w Centrum Obliczeniowym, w Instytucie Automatyki, na uniwersytecie. Wszystkie elementy trzeba było potem zwrócić. Ale na razie zostały rozstawione w pokoju, a w tym całym dziwacznym zestawie zamiast procesora "Elektroniki-60" uruchomiono pierwszego Kronosa, który chwilowo nie miał jeszcze nazwy.

Historia nazwy jest małą tajemnicą twórców. Powiemy tyle, że nazwano procesor na cześć greckiego tytana Kronosa, który - niestety! – pożerał własne dzieci i został strącony do Tartaru przez swego syna Zeusa. Później Zeusowi zrobiło się żal staruszka, więc skierował go na "wyspy szczęśliwości" w charakterze króla. Stąd też w wyobrażeniach starożytnych Greków "królestwo Kronosa" kojarzyło się z legendarnym "złotym wiekiem". Będąc początkowo bogiem żniw, Kronos był przedstawiany jako starzec z sierpem. Pojmowanie Kronosa jako boga czasu wykształciło się później, w wyniku podobieństwa brzmienia jego imienia i greckiego słowa "chronos" oznaczającego czas.

Jest cały szereg zabawnych historii z życiorysu Kronosa. Na przykład, by zakończyć spory i dyskusje, wszyscy złożyli podpisy pod zatwierdzonym wariantem architektury procesora. Zgodnie z nim maszyna miała być 16-bitowa (a chciałoby się mieć 32-bitową). I dosłownie po paru dniach, w nocy, Żenia Tarasow wymyślił, jak zrobić 32-bitowy procesor. W ciągu tej samej nocy przerobił cały system rozkazów. A gdy powiedział o tym przez telefon Dimie Kuzniecowowi, ten jedynie słabo zaoponował: – No przecież umawialiśmy się...

Na początku 1985 roku studenci dostali pierwszy pokój w Centrum Obliczeniowym SOAN ZSRR i od tego momentu wszystkie prace nad Kronosem prowadzono tutaj, w Centrum, przy poparciu ze strony Aleksandra Marczuka. Tu zrodzili się bracia pierwszego Kronosa: Kronos 2, Kronos 2.2, Kronos 2.5, Kronos 2.6. Trzy ostatnie wersje znane są obecnie jako rodzina procesorów KRONOS. Tu zaczęli pracę po ukończeniu studiów twórcy Kronosów i sformowała się grupa "Kronos" w obecnym składzie: Dmitrij Kuzniecow, Aleksiej Niedoria, Jewgienij Tarasow i Władimir Filippow. Oni stanowią trzon, wokół którego zgrupowało się jeszcze czternastu znawców i miłośników Kronosa, w większości – studentów uniwersytetu.

System operacyjny

Wszystkie procesory rodziny działają pod kontrolą systemu operacyjnego Excelsior. Na pytanie, dlaczego nie skorzystali z jakiegoś gotowego popularnego systemu, autorzy odpowiadają, że potrzebny był im system, odpowiadający ich koncepcji systemu operacyjnego, którą można wyrazić w kilku słowach tak: otwartość, modułowość, jednolitość wyników, wygoda interfejsów. Poza tym system nie powinien być skończony, zamknięty raz na zawsze, lecz rozwojowy, żywy. Bazowanie zaś od początku na produkcie programowym, którego autorzy mieli na myśli zupełnie inne zadania, nie jest rozsądne. Ponadto jeżeli coś zostało już zrobione, to znaczy, że było robione wczoraj i choćby dlatego nie nadaje się dzisiaj.



rys. Piotr Kakiet

Kronos a MARS

W 1985 r. zorganizowano tymczasowy zespół naukowo-techniczny "Start" pod kierownictwem Wadima Kotowa. Celem pracy zespołu był MARS – Modułowe Asynchroniczne Systemy Rozwojowe, praca zakończyła się sukcesem w 1988 r.

Prawie wszyscy specjaliści od Kronosa wzięli udział w pracach "Startu", konkretnie w tej części projektu, która nosiła nazwę MARS-T (T – jak transputer).

MARS-T jest realizacją takiego modelu obliczeń, gdzie zadanie dzielone jest na szereg procesów, które mogą być wykonywane na różnych procesorach. Sprzętowo jest to system wieloprocesorowy, którego elementy stanowią procesory Kronos 2.6, połączone za pomocą specjalnych urządzeń łączności międzyprocesorowej gwarantujących synchronizację procesów.

Jaka z niego korzyść?

Jak widać z krótkiego rysu historycznego, twórcy Kronosa chcieli zbudować maszynę przede wszystkim dla siebie. Wyszło tak, że Kronos stał się nie tylko celem pracy, ale i narzędziem do osiągnięcia tego celu. Całe oprogramowanie Kronosa pisane było na Kronosie, w języku Modula-2. Stąd pierwsze i podstawowe jego zastosowanie: opracowanie i realizacja rozwojowego oprogramowania. Rozwojowego – czyli takiego, które można łatwo przebudowywać i uzupełniać. I służy temu główna, naszym zdaniem, cecha Kronosa: programmability czyli łatwość programowania.

Jeśli dodamy do tego dużą przestrzeń adresową (do 8 gigabajtów w wersji 2.6) i niezłą wydajność oraz monitor graficzny, to mamy sprzęt do realizacji i funkcjonowania rozmaitych systemów CAD czyli work-station dla konstruktora.

Z kolei doświadczenia zespołu "Start" wykazały możliwość wykorzystywania procesorów z rodziny Kronos w rozbudowanych zestawach wieloprocesorowych.

Poza tym Kronos można zastosować w rozmaitych problemach obliczeniowych – są już przykłady takich zastosowań.

Czym właściwie są procesory?

Obecnie rodzina 2.X składa się z trzech wersji: 2.2, 2.5, 2.6. Wszystkie procesory mają ten sam system rozkazów, są całkowicie zgodne

programowo i różnią się jedynie wewnętrzną budową funkcjonalną, szybkością i wykonaniem konstrukcyjnym.

Logiką funkcjonowania wszystkich bloków procesora zawiaduje blok sterowania mikroprogramowego. Dwie szyny danych łączą jednostkę arytmetyczno-logiczną, blok rejestrów, szybki stos sprzętowy o pojemności 7 słów, licznik rozkazów i urządzenia wejścia/wyjścia.

Dwuszynowa struktura wewnętrzna procesorów pozwala na wykonywanie operacji binarnych na stosie (dodawanie, odejmowanie, logiczne AND, OR i in.) w jednym takcie. W ten sposób w jednym takcie wykonywana jest większość rozkazów, co odpowiada podstawowym założeniom architektury RISC. Sterowanie mikroprogramowe upraszcza budowę procesorów i daje możliwość realizacji skomplikowanych rozkazów typu wywołania procedury.

Wszystkie zespoły procesorów wykonano na bazie układów TTL (produkcji radzieckiej) z serii 155, 531, 1802, 1804, 589, 556.

Rodzina KRONOS: różnice i podobieństwa

Procesor	Kronos 2.2	Kronos 2.5	Kronos 2.6
Podstawa	Elektronika-60	Intel	Euromechanika
Ilość płyt	1	2	2[2]
Szyna	Q-bus 22	Multibus-1	lokalna[3]
Wielkość mikrorozkazu (bity)	56	64	64
Pojemność mikroprogramów (kB)	2	2	4
Bezpośrednio adresowana pamięć	4 MB	2,5 MB[1]	8 GB
Częstotliwość taktowania (MHz)	4	3	3
Liczba prostych operacji na stosie (mln/s)	0,6	1	1,5

1) Ważną cechą wyróżniającą Kronosa 2.5 jest obecność lokalnej pamięci o pojemności 0,5-2 MB, w zależności od zastosowanych układów. Reszta pamięci – na szynie Multibus-1 (do 1 MB).

2) Zestaw-minimum składa się z: płyty głównej (ALU, stos, rejestry), płyty sterowania mikroprogramowego, płyty pamięci lokalnej (0,5-2 MB), karty sterownika I/O.

3) Wszystkie urządzenia łączy lokalna synchroniczna 32 -bitowa szyna. Sam procesor nie jest uzależniony od konkretnej szyny I/O. Dopasowanie do konkretnej szyny odbywa się za pomocą odpowiedniej karty. Do szyny lokalnej mogą być dołączone układy pamięci, karta łączności międzyprocesorowej, sterownik sieci lokalnej i dysku twardego, układ pamięci kodu (w przypadku rozdzielnych układów kodu i danych), karta graficzna, zmiennoprzecinkowy koprocesor arytmetyczny oraz inne urządzenia rozszerzające możliwości procesora.

Tłumaczył z rosyjskiego:
Andrzej Popławski

Kurier

Rodzi się smok...

Z Thomasem Lamem, prezydentem firmy Flytech Technology Co. Ltd z Tajwanu, rozmawiają Tadeusz Wilczek i Tomasz Zieliński.

– Pańska firma istnieje pięć lat. Czy mógłby Pan podać nam nieco bliższych danych o niej?

– W 1984 roku, kiedy ją założyłem, zaczynaliśmy z czterema pracownikami i obrotem rzędu 1 miliona dolarów amerykańskich. Liczba pracowników rosła potem sukcesywnie i wynosiła w kolejnych latach: 16, 40, 65, by pod koniec ubiegłego roku osiągnąć poziom 75 osób. Nasz obrót w zeszłym roku wyniósł 13 milionów dolarów. Do tej pory wynajmowaliśmy lokal, w którym mieściła się nasza siedziba. Rok temu rozpoczęliśmy budowę własnej fabryki. W listopadzie zamierzamy się do niej wprowadzić, tak aby produkcję rozpocząć na początku 1990 roku.

– Dokąd sprzedaje pan swe produkty?

– W sumie eksportujemy 99% naszej produkcji, z tego do Europy ok. 50%, a 30% do USA i Kanady. Resztę wysyłamy do Australii, Nowej Zelandii, Hongkongu, Singapuru, Japonii i ostatnio także do Korei. W najbliższej przyszłości planujemy rozpocząć handel na rynku wschodnioeuropejskim.

– Jakie produkty pańskiej firmy są przedmiotem eksportu?

– Flytech wytwarza trzy grupy produktów. Pierwsza to płyty główne do komputerów PC/XT i PC/AT, karty graficzne, karty multi I/O oraz inne karty interfejsów przeznaczonych do komputerów standardu PC. Razem stanowi to około 50% naszej produkcji. Druga grupa (25 do 30% produkcji), to karty komunikacyjne, przede wszystkim modemy, karty sieciowe i emulacyjne terminali itp. Trzecia grupa, 20%, to urządzenia do sterowania przemysłowego. Reszta to sterowniki, urządzenia i karty wytwarzane na zamówienie klientów, zgodnie z dostarczonym projektem. Ostatnio np. nawiązałem kontakt z firmą Atari i dla niej będziemy wytwarzali niebawem specjalistyczny sterownik. Do tej pory produkowany był on w RFN, ale koszt produkcji był zbyt wysoki. Przedstawiciele firmy Atari zaproponowali mi wytwarzanie tego urządzenia na Tajwanie, co im pozwoli w zasadniczy sposób obniżyć koszty, a dla nas stanowić będzie intratny i interesujący kontakt.

– Firma Atari ze swoimi komputerami jest nieco na uboczu głównego nurtu PC.

– Niezupełnie, przecież Atari produkuje również komputery osobiste. Są one wprawdzie trochę inne niż PC, ale mają swój rynek. Nam jest wszystko jedno. My po prostu zapewniamy wysoką jakość produkcji, sprawny serwis i oczywiście konkurencyjną cenę.

– W Polsce znamy już trochę firmę Flytech, przede wszystkim z używanych u nas modemów. Ma pan już więc pewne kontakty z naszym krajem. Czy uważa pan, że mają one przyszłość?



– Myślę, że tak. Nasza firma nie jest jeszcze wystarczająco duża, aby bezpośrednio sprzedawać swe wyroby na całym świecie. Potrzebujemy autoryzowanych przedstawicieli. Chętnie nawiążemy kontakty z tymi, którzy chcieliby z nami współpracować.

– Jak to się stało, że założył pan tę firmę i tak ją rozwinął w ciągu niespełna 5 lat?

– Studiowałem na uniwersytecie elektronikę i technikę komputerową. Po uzyskaniu dyplomu postawiłem sobie jeden cel - zaprojektować jakiś układ elektroniczny, mający szansę zbytu, i rozpocząć jego produkcję. Przez pięć lat pracowałem w pewnej firmie i w końcu zdecydowałem, że dojrzałem do tego, aby rozpocząć samodzielną działalność. Założyłem więc swoją firmę.

– Na Tajwanie są setki firm komputerowych. Jak to jest możliwe, że, w sumie, w tak niewielkim kraju powstaje tak dużo firm?

– Tajwan to inny kraj, tam ludzie myślą inaczej. A małe firmy dobrze sobie radzą, ponieważ mają małe narzuty. Z czasem jednak łączą się one w większe przedsiębiorstwa. Prawidłowość jest więc taka, że w miarę umacniania się i wzrostu te małe stopniowo znikają.

– A pańska firma zalicza się do....

– ... firm średnich. Ale, jak już wspomniałem, rozwijamy się, budujemy własną fabrykę.

– Jakie plany produkcyjne na przyszłość?

– Dzisiaj nasza produkcja jest nastawiona na PC. Jednocześnie rośnie nasze zainteresowanie produktami z zakresu telekomunikacji i sterowania procesami przemysłowymi. Mamy też ciekawy kontrakt z pewną firmą amerykańską na dostawę karty syntetyzującej głos ludzki.

– Czy nadal projektuje pan poszczególne wyroby osobiście?

– Do niedawna jeszcze tak. Zaprojektowałem około 65% naszych produktów. Przekazałem jednak pałeczkę zgranemu zespołowi specjalistów, a sam zająłem się kierowaniem firmą. Jeżdżę po wystawach, spotykam się z naszymi głównymi kontrahentami, myślę jak rozwijać firmę. W kwietniu otworzyliśmy biuro we Frankfurcie. Stworzenie dobrego serwisu jest naszym krokiem w kierunku europejskich klientów. Jednym z zadań tego biura będzie udział w wystawach na terenie całej Europy, także w Polsce, ZSRR i innych krajach socjalistycznych.

– Życzymy zatem sukcesów na rynkach wschodnioeuropejskich.



rys. Piotr Kakiet

Kurier

Komputeryzujemy się

Komputer i szkoła, komputer i uczenie się to pojęcia od dawna związane ze sobą, a w różnych stronach świata tak ściśle, że wręcz są synonimami. Niestety nie możemy tego powiedzieć o naszym kraju. A szkoda. Temat ten jednak często gości na łamach polskiej prasy.

Pesymistyczną diagnozę naszej "szkolnej" komputeryzacji stawia "Dziennik Zachodni".

"Rewolucja mikrokomputerowa dotarła – jak wiadomo – również i do nas.(...) Od pewnego czasu ta rewolucja puka i do drzwi szkolnych. I tu sytuacja budzić musi autentyczny niepokój, gdyż drzwi te – z różnych względów – uchylają się z wielkimi oporami. Po pierwsze są kłopoty ze sprzętem. Komputery są dość drogie – mówi się. Ale to niezupełnie prawda (...) Tak więc mówienie o kosztach jest jednak wymówką, obroną przed nowoczesnością. Gorzej z tym, że do dziś nie ma masowej produkcji jednolitego typu mikrokomputera szkolnego.(...) Najważniejsza sprawa jest jednak z czym innym. Otóż dziś komputer – jeśli już nawet trafia do szkoły – budzi ogromne zakłopotanie pedagogów i dyrekcji. Nauczyciele nie mają o nim na ogół zielonego pojęcia. Jeśli zaś mają – to wiedza komputerowa nie jest uwzględniana w programach (...) Wygodniej więc i taniej – jak w pewnej znanej mi szkole – otrzymawszy komputer zamknąć go w pancernej kasie i nie robić z nim kompletnie nic... Powiedzmy sobie szczerze: jeśli wszystko to będzie wyglądało tak jak dotychczas – czeka nas niemiła perspektywa analfabetyzmu komputerowego.To zaś oznacza cofnięcie Polski do czwartego, piątego czy któregoś tam świata bez względu na bogactwo kraju, którego oczywiście też dziś nie ma. Lecz jeśli nawet zdarzy się u nas "cud gospodarczy", to i tak bez komputeryzacji nie wyjedziemy z dołka cywilizacyjnego. Trzeba temu przeciwdziałać. Konieczne jest przede wszystkim szybkie wprowadzenie informatyki do wszystkich szkół i specjalności nauczycielskich. Informatyka musi się stać tam przedmiotem tak samo ważnym jak psychologia rozwojowa czy pedagogika. Po drugie, czas zakończyć spory o programy szkolne. (...) Wiedza komputerowa musi jak najszybciej – w 2-3 latach – stać się obowiązkowa w szkołach średnich, zaś najdalej w ciągu kilku lat wejść do programów szkół podstawowych. Musi być także podjęta naprawdę masowa produkcja mikrokomputera szkolnego, liczona setkami tysięcy sztuk rocznie."

Nic dodać nic ująć. Zgadzamy się z poglądami autora i podpisujemy obydwiema rękami pod myślą zawartą w zdaniu kończącym tekst.

"Bez komputera – traktowanego jak przedłużenie własnej ręki (czy może lepiej: "wzmocnienie głowy") – już wkrótce nie da się żyć przyzwoicie nigdzie. Chyba że chcemy za niewiele lat zapewnić Polakom światową wyłączność na obsługę... łopaty."

* * *

O tym jak wygląda szkolna rzeczywistość z pozycji kuratorium, donosi krakowski "Dziennik Polski".



rys. Piotr Kakiet

"Pan Henryk Białek z krakowskiego kuratorium, wizytator ds. komputeryzacji szkół mówi nam, że do tej pory specjalnej puli na sprzęt komputerowy nie było. Ministerstwo Edukacji Narodowej posiada fundusz przeznaczony tylko dla szkół średnich. Z tego funduszu korzystając planowano na 1988 r. stworzenie 18 pracowni po 10 mln zł każda. Czemu tak drogo? Bowiem w ramach popierania gospodarki narodowej Ministerstwo Edukacji miało przykazane, by dla szkół kupować wyłącznie Juniory, które sprzedawane jako sieć kosztują ok. 10 mln zł. Oczywiście są inne komputery, o kilka klas lepsze i kilka milionów tańsze, ale przecież my mamy swoje! Co z tego, że kiepskie? (...) Od jakiegoś czasu istnieje zarządzenie, na mocy którego szkoła z funduszu inwestycyjnego może kupić sprzęt komputerowy i jeśli będzie to sprzęt typu Timex lub Spectravideo (sztuka ok. 140 tys. zł!) nabyty w Składnicy Harcerskiej, to kuratorium może ten wydatek zrefundować. Ta decyzja wydaje się dość zasadna, bowiem wymieniony sprzęt jest w miarę tani, bardzo dobry, a kupowanie przez wszystkie szkoły tego samego typu komputerów ma na celu standaryzację, a więc także możliwość wymiany. Szkoły radzą sobie jak mogą. Szukają bogatych sponsorów. (...) Cóż stąd, kiedy w dzisiejszych czasach przedsiębiorstwu nie opłaca się podarować szkole komputera, bowiem obciąża się go bardzo wysokimi, karnymi podatkami za "koszty nieuzasadnione". Wszędzie na świecie ofiarodawca zwalniany jest od podatków, u nas odwrotnie – karany jest za to, że chce pomóc. (...) W Zespole Szkół Elektrycznych Nr 1 w Krakowie pracownię komputerową kompletuje się od 1986 r. Dziś jest to bardzo dobrze działająca pracownia, wyposażona w 16 egzemplarzy komputerów, lecz większość to właśnie ELWRO Junior. Nauczyciele informatycy mówią, że gdyby mogli, kupiliby Amstrady lub IBM-y. - W szkole połączenie w sieć nie jest konieczne, bo nauczyciel z łatwością może przejść po klasie i sprawdzić wszystkie monitory – dorzucają. – Wszystko jest tańsze od ELWRO."

"Sztandar Młodych" również zamieszcza zwierzenia wizytatora.

"Ostatnio pęd do zdobywania sprzętu w naszym województwie trochę przygasł – informuje Zenon Bukowski, wizytator- metodyk ds. informatyki w Radomskiem. – Wiele naszych szkół nie ubiega się o komputery, bo twierdzą, że są im niepotrzebne.

Nic dziwnego, bo sprzęt przyszedł nie wydarzony. Rzadko która dostawa nie kwalifikuje się zaraz po rozpakowaniu do zwrotu lub naprawy. Na razie za serwis gwarancyjny płaci ministerstwo. Przykładowo naprawa jednego klawisza kosztuje ok. 40 tys. Kto będzie jednak płacił, gdy skończy się gwarancja? Kuratoria pieniędzy nie mają. Ubodzy jesteśmy również w edukacyjne oprogramowanie. Opracowany przez Ośrodek Badawczo-Rozwojowy Pomocy Naukowych "Katalog Oprogramowania Dydaktycznego" nie cieszy się dużą popularnością. Dlatego większość nauczycieli pisze programy samodzielnie."

Oj, nie jest łatwe życie wizytatora ds. komputeryzacji szkół. Pieniędzy nie ma. Programów też nie ma, a bez nich nawet najlepszy sprzęt jest bezużyteczny. Komputerów szkoły nie chcą. Gdy się znajdzie jakiś "dobry wujek" i będzie chciał zafundować komputer, to się przestraszy "kosztów nieuzasadnionych".Oj, kiepsko. Nie zazdrościmy też uczniom Zespołu Szkół Elektrycznych nauczycieli. Jeśli dla nich sieć jest równoważna bieganiu po klasie i spoglądaniu na uczniowskie monitory, to może wystarczy im kreda i tablica. Proste to i znane od dawna, i na pewno tańsze od "Elwro".

* * *

Tygodnik "Odrodzenie" zajął się sprawą realizacji programu powszechnej edukacji w zakresie wiedzy informatycznej w latach 1986-1990. Realizacja programu, powstałego w marcu 1986 roku, jest poważnie zagrożona. Taką opinię wyraziła Komisja Nauki i Postępu Cywilizacyjnego Warszawskiej Rady PRON. Oto jej fragment:

"Program edukacji informatycznej nie zostanie zrealizowany, jeśli nie zmieni się program na poziomie średnim i wyższym oraz nie zwiększy się dostęp do literatury fachowej, materiałów szkoleniowych i oprogramowania na użytek uczelni i szkolnictwa średniego.

Nadal nie opracowano rozwiązań prawnych dotyczących rozpowszechniania i ochrony praw autorskich. Brak tych regulacji uniemożliwia powstanie normalnie funkcjonującego rynku oprogramowania, rzutuje negatywnie na międzynarodowe kontakty naukowe, utrudnia dostęp do powstających w świecie nowoczesnych programów. Członkowie Komisji zwracają również uwagę na niską jakość produkowanych przez ELWRO komputerów szkolnych i urządzeń peryferyjnych."

To już znamy. Znowu mamy bardzo dobry program i bardzo złą realizację.

* * *

"Trybuna Robotnicza" przedstawia sposób na komputeryzację w szkole na przykładzie Liceum Ogólnokształcącego w Żorach.

"W połowie lat osiemdziesiątych fala zainteresowania komputerami, ich możliwościami technicznymi dotarła i do naszego miasta – wspomina dyrektor LO mgr Andrzej Leśnik. Z tego, że nie ominie ona szkoły, zdawano sobie sprawę. (...) 4 komputery typu Spectrum stały się zalążkiem pracowni komputerowej, która obecnie wygląda im-

ponująco: wspomniane komputery typu Spectrum, 10 komputerów Elwro Junior połączonych w sieć i nasza chluba – 2 komputery typu IBM (z twardym dyskiem i bez niego). (...) M.in. oprócz pomocy w procesie dydaktycznym komputery mogą być wykorzystane do celów administracyjno-zarządzających w szkole – obecnie m.in. opracowywany jest przez uczniów program dla tzw. gospodarki magazynowej w szkole. W przyszłości i inne dane administracyjne (poprzez opracowanie programów) zostaną wprowadzone do pamięci komputera. (...) Programowanie nie jest podstawą pracy z młodzieżą. Tym zajmują się ci, którzy dobrze poznali to urządzenie i co nie mniej ważne, chcą to robić. Główny cel pracowni to pokazanie młodzieży, w jaki sposób i do jakich celów może być wykorzystany komputer. (...)

Oczywiście zajmujemy się – mówi nauczyciel Adam Herman – także tworzeniem programów. W oparciu o taki sprzęt robimy m.in. profesjonalne programy. Najczęściej tworzone są one w wyniku zlecenia ich Młodzieżowej Spółdzielni Pracy "PRO.S.I.A.K", czyli zespołowi utworzonemu w oparciu o uczniów żorskiego liceum. Praca w niej to dla młodzieży nie tylko nowe, ciekawe tematy do realizacji, ale co tu ukrywać także i pieniądze. (...) wykonaliśmy m.in. system zarządzania kartoteką opieki społecznej (dla ZOZ-u w Żorach), indywidualne programy komputerowe dla rzemieślników. (...) Pieniądze uzyskiwane z wykonanych prac zasilają kieszeń wykonawców oraz częściowo budżet szkoły. Przeznaczone są także na bieżące utrzymanie pracowni komputerowej."

Również w "Sztandarze Młodych" czytamy:

"Niedawno w Krakowie wprowadzono kilka komputerów w jednej ze szkół specjalnych. W pierwszej kolejności dostęp do maszyny miały dzieci z wadami wymowy i szczególnie poważnymi zaburzeniami nerwowymi. Już po miesiącu efekt był zaskakujący. Te same dzieci, które wcześniej z trudem przechodziły z klasy do klasy, zaczynały nawet lubić szkołę. Komputer okazał się w tym przypadku czymś więcej niż tylko pustą maszyną."

Te dwa przytoczone przykłady pokazują, że jak się chce to można. Wierzymy, że jest więcej takich szkół, gdzie nie czeka się na lepsze czasy siedząc z założonymi rękami.

Krzysztof Matey

\Kurier\

Stanisław Marek Królak



Terminator terminologiczny [25]

Wakacje, pora rozkosznego lenistwa nad morzem i jeziorami, zobowiązują do nieco lżejszego spojrzenia na tematy poruszane w tej rubryce. Dlatego ten felieton poświęcam sprawom prostym, niekiedy wręcz oczywistym, i traktuję jako powtórkę przerobionego materiału.

Co oznacza *handshaking* w informatyce, wiemy zapewne wszyscy i pojawienie się tego słowa w tekście nie wywołuje nieporozumień. Co jednak powiedzieć o autorze zwrotu *...w metodzie handshakingu mamy...?* Przecież od dawna powszechnie przyjętym tłumaczeniem terminu angielskiego jest wyrażenie ***przesyłanie z potwierdzeniem***. Nam to nie przeszkadza, mówią pożal się Boże specjaliści, a poza tym są tu trzy słowa, a nie jedno. I w tym momencie jestem bezradny, bo idąc za Herbertem mogę wysunąć tylko jeden argument: że jest to kwestia smaku. Ale czy zrozumieją? A co to jest smak? – mogę usłyszeć. Albo jeszcze lepiej: A kto to jest Herbert?

Podobnie się ma sprawa ze słowem *display*. Termin nie jest nowy i jego odpowiednikami są ***wskaźnik***, ***wyświetlacz*** ewentualnie ***monitor***. Nie ma więc żadnego powodu, aby pisać *display*, a tym bardziej *dysplej* (w swoim dorobku tropiciela kretynizmów językowych mam na razie jeden przypadek takiej pisowni).

Nieco bardziej skomplikowana jest sprawa, jeśli chodzi o wyrażenie *interface*. Otóż w elektronice i informatyce często używane jest pojęcie *układ sprzęgający*. W niektórych publikacjach spotkać można także określenie ***sprzęg***. Tymczasem w publikacjach popularnych używa się słowa interfejs – tak też jest w naszym piśmie. Na pewno nie jest to rozwiązanie najszczęśliwsze, a wzięło się po prostu stąd, że nie byliśmy przygotowani do gwałtownego rozwoju mikrokomputerów – baza wyprzedziła nadbudowę. Dyskusje w naszej redakcji również nie przyniosły rozstrzygnięcia. Sympatyczna propozycja: *międzymordzie* miała wielu przeciwników. Myślę, że wyrażenie ***układ sprzęgający*** jest do przyjęcia. Ale co zrobić wówczas na przykład z programem określanym jako *interfejs użytkownika*? Czy program może być układem? Jeśli nie, to może rozdzielić pojęcia i pisać ***program sprzęgający?***

Na koniec dwa słowa: *controller* i *joystick*. W pierwszym przypadku sprawa jest banalna – określenie kontroler jest błędem językowym i nie może być przyjęte w żadnym przypadku. Przypominam więc, że mamy ***sterownik*** i tylko sterownik. Kontroler to co najwyżej osobnik, który okazuje "blachę" i żąda biletu lub pięciu tysięcy.

Angielskie *joystick* – pisane zwykle *joy-stick* – związane z lotnictwem, miało swój odpowiednik w polskim ***drążek sterowy***, określany w żargonie pilotów również jako orczyk. W popularnej literaturze mikrokomputerowej spotyka się różne pisownie – najczęściej *joystick*, chociaż trafić czasem można na kuriozalne *dżojstyk*. Coraz częściej pojawia się też ostatnio określenie ***manipulator***. Nie jest ono może bardzo eleganckie, ale niesie jakąś treść i nie kłóci się z duchem języka polskiego. Przypuszczam, że właśnie tę formę będziecie Państwo najczęściej spotykać na naszych łamach.

\Kurier\



Nowości

Z Zachodu...

Brytyjska firma ICI wydobyła z zeszłorocznych targowych "szpargałów" opracowany przez siebie nośnik danych, tzw. papier cyfrowy, i prezentowała jako nowość także na tegorocznym CeBIT-cie. Cena jednak pozostała bez zmian – jeden megabit kosztuje nadal feniga.

Już dziś 1 MB przechowywany na papierze cyfrowym kosztuje tylko 1 feniga. Nowy nośnik dostępny jest w postaci dyskietek lub zwykłej taśmy. Na taśmie długości 2400 stóp (ok. 800 m) i tworzącej szpulę o średnicy 26 cm można zmagazynować przy obecnym stanie techniki 600 gigabajtów. Podobnie jak w przypadku pamięci ROM, raz zapisanych danych nie można skasować.

Oprócz korzystnego stosunku ceny do możliwości nowy nośnik cechuje także długa żywotność – według informacji producenta trwałość zarchiwizowanych danych wynosi obecnie 15 lat i ulegnie przedłużeniu o dalsze pięć.

Przewidywanymi obszarami wykorzystania nowego nośnika byłyby, obok magazynowania danych geofizycznych, meteorologicznych i satelitarnych, także systemy ekspertowe i dokumentacja medyczna.

W związku z pojawieniem się papieru cyfrowego zapowiedziane zostały dwie optyczne stacje dysków elastycznych, które firma **ICI** opracowała we współpracy z Bernoulli Optical Systems Corporation (BOSCO) z Colorado. Oba urządzenia działają przy wykorzystaniu efektu Bernoulliego, a na rynku pojawią się w przyszłym roku.

Już w tym roku jednakże ukazać ma się specjalny *streamer* o pojemności jednego terabajta, opracowany przez kanadyjską firmę Creo Products Inc. z Vancouver. Według informacji firmy brytyjskiej, dostęp do danych w obrębie jednego terabajta wynosi przeciętnie 28 sekund.

(PW)

* * *

Mimo iż od prezentacji **MS-DOS**-a 4.0 upłynęło niewiele czasu, Microsoft zapowiedział już **wersję 4.01** zgodną z PC - DOS 4.01 wspierającą standard EMS.

(PW)

* * *

Framework III LAN zapowiedziany został przez firmę Ashton-Tate. Jak wskazuje nazwa, będzie to wersja sieciowa współpracująca z Novell Advanced i SFT Netware/286, Token-Ring, IBM-PC-LAN oraz 3ComNet 3+.

(PW)

* * *

W numerze marcowym "Chipa" znajdujemy ciekawe informacje o nowościach programowych prosto z USA, które mają pojawić się wkrótce w Europie. Zostały one zanalizowane w różnych grupach zastosowań.

I tak w grupie procesorów tekstu największe nadzieje wzbudza zapowiedziana **wersja 5.0** programu **Word** firmy Microsoft. Nowa wersja ma się odznaczać większą szybkością oraz możliwością graficznego przedstawienia całej strony przygotowanej do druku (w pełni WYSIWYG). Będzie ona także umożliwiać redagowanie tekstu w formacie wieloszpaltowym oraz swobodne włączanie do niego grafiki. Każda zmiana w zbiorze zawierającym rysunek, który ma zostać zamieszczony w tekście, spowoduje także modyfikację tego rysunku w przetwarzanym dokumencie. Podobnie "inteligentnie" działać będzie również interface do programów kalkulacyjnych Lotus 123 i Excel. Uproszczone zostało także tworzenie i modyfikacja ramek z tekstem lub grafiką. Word 5.0 zgodny jest z systemem OS/2.

Wśród baz danych najbardziej godny uwagi wydaje się program do archiwizacji zdjęć **PCAlbum**. Służy on do przechowywania zdjęć wykonanych kamerą cyfrową wraz z ich opisami. Do przewidywanych zastosowań należy ewidencja zbiorów, a także tworzenie kartotek personalnych, np. w celu ochrony przed dostępem osób niepowołanych.

W grupie programów graficznych wyróżnia się **Corel Draw**. Oprócz tradycyjnych funkcji odre-

> 10

9 <

cznego rysowania i manipulowania tekstami umożliwia on automatyczną modyfikację rysunków wgrywanych poprzez skanner.

Z uwagi na wysoką cenę drukarek laserowych, zgodnych ze standardem Postscript, dużą popularnością cieszą się interpretery tego języka np. **Ultra Script PC** i **Go Script plus**. Za ich pomocą można drukować zbiory w formacie Postscript na zwykłych drukarkach laserowych, a nawet matrycowych.

Wśród pakietów testujących należy wymienić program **CheckIt**. Oprócz tradycyjnej diagnostyki systemu wyświetla on "zajęte" przerwania, co jest szczególnie przydatne przy montowaniu nowych kart i ustawianiu przerwań. Program oferuje także możliwość pomiaru częstotliwości zegara i czasu dostępu dysku twardego.

(pm)

...i ze Wschodu

"PC World" po rosyjsku

Nakładem radzieckiego wydawnictwa "Radio i Swjaz'" ukazał się pierwszy numer "W mire personalnych kompiuterow", czyli popularnego amerykańskiego miesięcznika komputerowego "PC World" w rosyjskiej wersji językowej. Pierwszy numer, w znacznej części wypełniony materiałami radzieckich autorów, zawiera również najciekawsze materiały z kilku różnych numerów oryginalnego "PC World".

Na uwagę zasługuje przede wszystkim historia komputerów klasy IBM PC, porównawcza analiza różnych wersji DOS-a, a z ciekawostek – materiał znanego radzieckiego szachisty Garri Kasparowa o tym, jak zakładał w Moskwie klub komputerowy *Komputer* (pisaliśmy o nim w numerze 9/88 naszego miesięcznika).

Według informacji uzyskanych przez nas w Moskwie, drugi numer periodyku (pierwszy ukazał się z datą ubiegłoroczną) spodziewany jest jeszcze w tym roku, podobnie jak radziecka edycja innego znanego pisma "PC Magazine".

(mc)

Dyskietki "made in USSR"

W Kijowie ruszyła pierwsza radziecka fabryka dyskietek. Produkowane tu dyski 5,25 cala na razie, jak nas poinformowali zaprzyjaźnieni komputerowcy z Rygi, bazują na materiałach zachodnich, więc ich jakość jest nienaganna. Według zapowiedzi sprzedawane będą dopiero w przyszłym roku. Znana też jest już ich cena – 10 rubli (1 rubel w transakcjach handlowych ma wartość 280 zł) za dyskietkę DS DD 80-ścieżkową.

Według zapewnień producenta już wkrótce spodziewać się należy również dyskietek 3,5-calowych.



Tak więc w naszym obozie elastyczne nośniki magnetyczne produkowane są już przez 3 kraje – Bułgarię, Węgry i ZSRR. A o zapowiadanych od paru lat dyskietkach z gorzowskiego Stilonu jakoś cicho.

(mc)

Montownie komputerów na Łotwie

Prasa szwedzka donosi o porozumieniu podpisanym przez jedną z firm skandynawskich i bliżej nie sprecyzowanym przedsiębiorstwem łotewskim (na jego ślad nie zdołali wpaść nawet informatycy z Uniwersytetu w Rydze) o uruchomieniu na terenie tej republiki montowni komputerów klasy PC. Gros produkcji przeznaczone ma być, według zapewnień tejże prasy, na eksport.

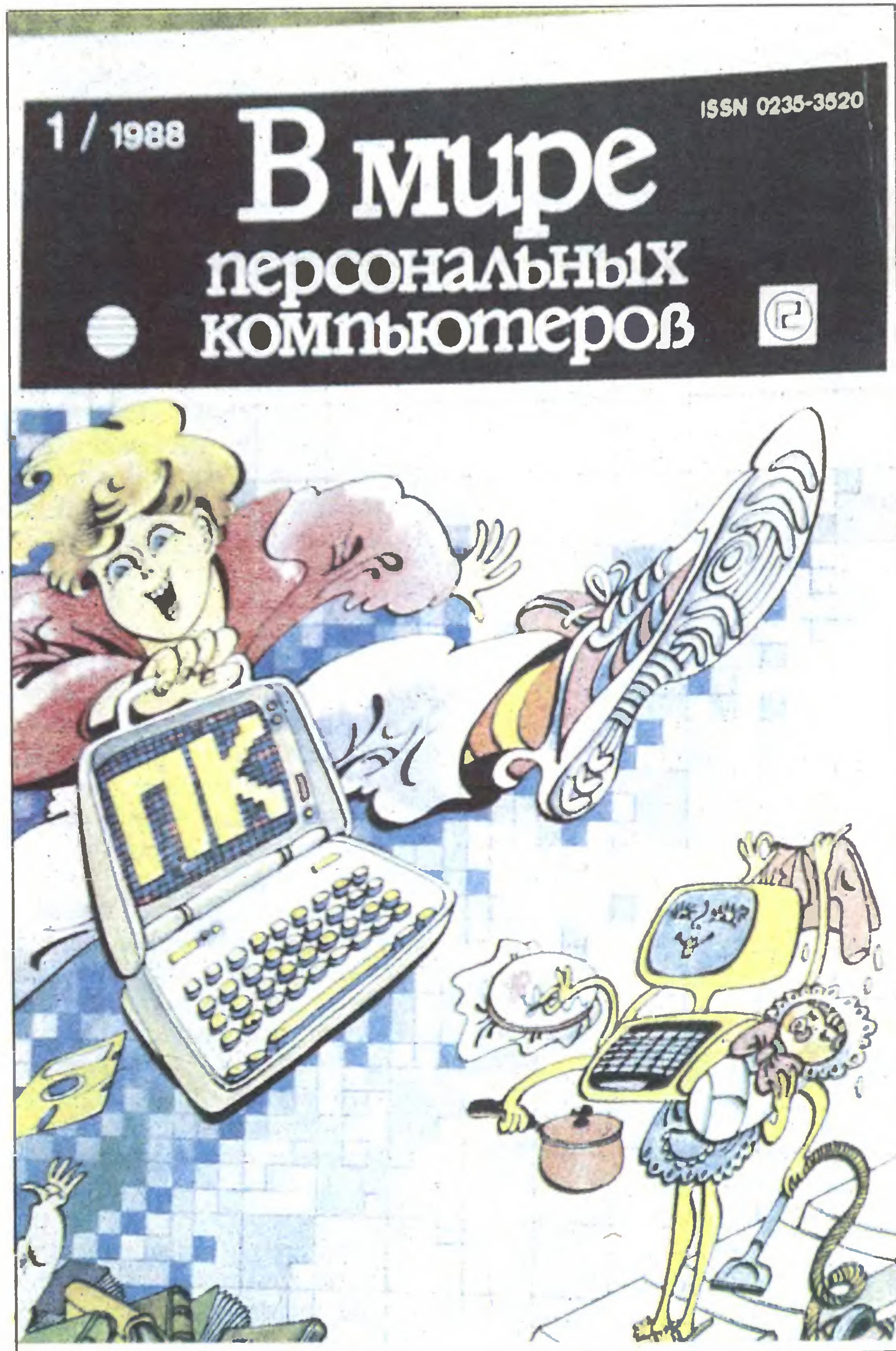
Jedna z warszawskich firm komputerowych zawarła również podobne porozumienie z przedsiębiorstwem grupującym największe kooperatywy (spółdzielnie) Rygi oraz znane w Związku Radzieckim zakłady RADIOTEHNIKA, produkujące aparaturę radiową i sprzęt nagłaśniający. Przewiduje ono uruchomienie w Rydze jeszcze w tym roku montowni komputerów, której docelowa "przepustowość" sięgnie 1000 maszyn rocznie. Jak na razie nierozwiązanym problemem pozostaje, bagatela, kwestia transferu zarobionych rubli do Polski.

(mc)

Po sprzęt do ...Wilna

Kooperatywa (spółdzielnia) działająca przy Instytucie Cybernetyki w Wilnie podpisała porozumienie z brytyjską firmą SPEC, dotyczące sprzedaży w ZSRR, za waluty wymienialne, sprzętu komputerowego po cenach konkurencyjnych nawet z dostawcami singapurskimi. Tak np. 12-megahercowe AT z twardym dyskiem 40 MB, kartą VGA, kolorowym monitorem "Multisync" i podtrzymywaczem napięcia (UPS) kosztować ma jedynie 2900 dolarów. Za 850 dolarów oferowane jest XT z kartą Hercules, monitorem monochromatycznym i twardym dyskiem 20 MB. Najtańsze drukarki – Shinwa – kosztować mają w Wilnie jedynie 170 dolarów (dla porównania w Berlinie Zachodnim za najtańszą drukarkę Stara – LC-10 – zapłacić trzeba 430 marek). Największą jednak atrakcją wydaje się zapowiedź sprzedaży drukarek laserowych RICOH za jedyne 1710 dolarów. Czyżby generalna zmiana kierunku naszych komputerowych zakupów?

(mc)



Wirusy w Moskwie

Wirusy wyraźnie dały się we znaki programistom radzieckim. Świadczy o tym duża liczba dobrych programów antywirusowych, które dotarły do naszej redakcji po kolejnych wojażach dziennikarzy po Kraju Rad. Większość z nich napisana została w różnych jednostkach organizacyjnych Akademii Nauk ZSRR. Autorami tych najlepszych, z których niekiedy zmuszeni jesteśmy korzystać, są m.in. Alexy Gusew (program VIRUS REMOVER), S. Andrejew i A. Hodujew (CONVIR) oraz Witalij Ładygin (program VIRUS DETECTOR, usuwający 2 typy wirusów ze zbiorów .COM i .EXE).

W naszym posiadaniu są również dwa bardzo dobre programy antywirusowe autorów czechosłowackich – Z. Hladka z Brna (program ANTIVIRUS PLUS) i Petra Horskiego, którego program VIROLOG nie tylko wskazuje zarażone zbiory, lecz również je leczy i uodparnia na działanie określonych wirusów, uniemożliwiając im dopisywanie się do zbiorów.

(mc)

Kurier

Postaci mikroświata

HEINZ NIXDORF

Swoją zawrotną karierę przemysłowca zaczął jeszcze na przełomie lat pięćdziesiątych i sześćdziesiątych, gdy słowo komputer symbolizowało jedynie postęp techniczny i adresowane było raczej do przyszłych pokoleń masowych użytkowników. W RFN nieliczna była wtedy grupa producentów sprzętu komputerowego, którzy wierzyli, że ich produkty będą niewyczerpanym źródłem zysku. Jednym z pierwszych, startującym wówczas, a jak się z czasem okazało najszybszym i zbierającym z tego tytułu największe profity, był Heinz Nixdorf.

Jak wielu ludzi z tej branży nie

NIXDORF
COMPUTER

zdążył ukończyć studiów na wydziale fizyki, a potem ekonomiki zakładów pracy. By zarobić podczas studiów na utrzymanie (pierwsze kroki w komputerowej branży stawiał w 1951 r.), pracował przez pół roku w komputerowym przedsiębiorstwie Remington jako asystent szefa oddziału konstrukcyjnego. Zdobył tam zasób wiedzy praktycznej, która umożliwiła mu samodzielne konstruowanie elektronicznych maszyn liczących. Przerwał więc studia, założył własne przedsiębiorstwo i zaczął szukać klientów. Pierwszy zlecił mu skonstruowanie komputera wypłacając zaliczkę w wysokości 30000 marek, oddając sekretarkę oraz piwnicę jako pomieszczenie do pracy. Tak więc swą karierę zaczął Nixdorf od samego dołu – i w przenośni, i jak najbardziej dosłownie.

Za pierwszym zleceniem przyszły następne, jednak prawdziwy przełom w dziejach potężnej dziś firmy nastąpił w 1965 roku, gdy na targach hanowerskich zaprezentowała ona swój mini-komputerek "Nixdorf 820". Aż do połowy lat siedemdziesiątych maszyny te były w RFN stosunkowo powszechnie stosowane jako komputery biurowe, elektroniczne maszyny do fakturowania lub jako terminale. Od tego czasu obroty przedsiębiorstwa nieustannie rosły.

Największy sukces westfaldzki przedsiębiorca zapisał na swym koncie w 1979 roku, wtedy to Deutsche Bank zgodził się zainwestować 200 mln marek kupując od Nixdorfa 25 procent udziałów jego Towarzystwa Akcyjnego NIX-



DORF COMPUTER. Warunki tej transakcji nie mają sobie równych w historii gospodarki zachodnioniemieckiej: otóż Nixdorf zastrzegł sobie prawo do zadecydowania w ciągu 4 lat, czy nabyte udziały bank będzie mógł sprzedać na giełdzie. Oczywiście nie zgodził się i suma 200 mln marek stanowiła jedynie bezpłatny kredyt.

Dzisiaj Towarzystwo Akcyjne NIXDORF COMPUTER z siedzibą w Paderborn może się pochwalić zainstalowaniem u swoich klientów blisko 200000 komputerów różnej klasy. W 11 zakładach krajowych i 33 fabrykach za granicą zatrudnia ono ok. 20000 ludzi. 7300 członków załogi posiada 5,8 milionów marek udziałów w kapitale zakładowym. W zarządzie przedsiębiorstwa miejsce Heinza Nixdorfa, który zmarł w ubiegłym roku, zajął jego wieloletni współpracownik i dotychczasowy zastępca Klaus Luft. O nim jednak innym razem.

(mc)



rys. Piotr Kakiet



Kurier

Czytaj!

Keith L. Clark, Frank G. McCabe "Micro-Prolog", z angielskiego przełożył Paweł Luboński, WNT 1988, wyd. I, 12 700 + 300 egz., 367 str., 700 zł, seria "Biblioteka Inżynierii Oprogramowania".

Wydaje się, że Prolog a tym bardziej micro-Prolog nie są jeszcze popularnymi językami programowania w naszym kraju. Tym niemniej warto odnotować pojawienie się omawianej książki – przede wszystkim ze względu na jej walory poznawcze. Ci z Państwa, którzy zetknęli się już z Prologiem, znajdą w niej usystematyzowane omówienie zasad programowania w tym języku; inni, którzy dopiero chcieliby go poznać, mogą potraktować książkę także jako wprowadzenie do rodziny języków programowania w logice (Prolog to skrót od "Programming in logic"). Wydana w 1984 roku w Anglii została już przetłumaczona na 18 języków. Atrakcyjność książki wynika między innymi z podejścia do programowania od strony logiki formalnej, co powoduje, iż od Czytelnika nie wymaga się wiedzy z informatyki. Więcej, Autorzy w tej metodzie upatrują perspektyw rozwoju języków programowania i nowych możliwości. Powołują się tu na prowadzone prace badawcze, np. program badawczy "Logika jako język komputerowy dla dzieci".

Książkę podzielono na cztery podstawowe części. W pierwszej wprowadza się micro-Prolog na przykładzie zakładania bazy danych i wyszukiwania w niej informacji. Łatwość, z jaką można zbudować taką bazę i korzystać z niej, stanowi jedną z podstawowych zalet języka. Omówiono też arytmetykę Prologu (jest ona zupełnie inna niż arytmetyka w konwencjonalnych językach programowania), wprowadzono reguły i definicje rekurencyjne oraz pojęcie listy i sposoby strukturalizowania informacji za pomocą list.

W części drugiej na przykładzie micro-Prologu przedstawiono język programowania w logice, po to, by w części następnej przedstawić standardową składnię.

Książkę kończą rozdziały, w których kilku autorów demonstruje niektóre zastosowania micro-Prologu.

* * *

Marian Łakomy "Word 3.1", WNT 1988, wyd. I, 19 700 + 300 egz., 92 str., 300 zł, seria "Podręczna Pamięć Programisty".

Kolejna ściągawka z rozpoczętej niedawno serii. Tym razem rzecz dotyczy popularnego edytora firmy Microsoft. Samouczek jest zgrabnie skomponowany i napisany, zawiera omówienie podstawowych zagadnień i problemów, wykaz poleceń dostępnych z menu oraz funkcji z klawiatury. Zabrakło natomiast przykładów, które moim zdaniem są niezbędne w tego typu książeczkach.

S.M.K.

Kurier

Krzysztof Matey

Mikro Historicus

O zakończeniu konkursu Mikro Historicus na program komputerowy o tematyce historycznej już pisaliśmy. W najbliższych numerach będziemy przedstawiać nagrodzone programy.

Redakcja

Program: *Miasto Historii*
Scenariusz: *Jacek P. Sobierajski*
Programista: *Jacek Jurczyk*
Komputer: *ZX Spectrum*

Na swojej drodze prowadzącej do siedziby Klio napotkałeś dziwne, nieznane miasto. Mimo obaw co cię czeka wkraczasz w jego mury. Pełno tu ponurych ulic, zaułków i ciemnych bram. Przebywasz w nim już kilka godzin i nie spotkałeś żadnego człowieka. Rozglądasz się wokoło, nie ma "żywej duszy", tylko nazwy ulic jakieś znajome. Tak, już wiesz, wszystkie związane są z historią Polski. Czy twoja wiedza pomoże ci wydostać się z labiryntu ulic? Jeśli chcesz spróbować, musisz zagrać w "Miasto Historii".

"Miasto Historii" to test połączony z grą zręcznościową. Gracz wciela się w postać tropiciela historii poruszającego się po dziwnym mieście. Wszystko w nim, ulice, zaułki, bramy, place, pomniki są związane z historią Polski. Zadaniem gracza jest przejście przez miasto i wydostanie się z niego w określonym czasie. Niestety, nie można przez nie po prostu przebiec. Przebycie miasta wymaga odpowiedzi na rozliczne pytania. Tylko prawidłowe umożliwiają graczowi posuwanie się naprzód.

Grę rozpoczynamy wyborem sposobu sterowania. Mamy do wyboru klawiaturę lub joystick. Ukazuje się wtedy na ekranie monitora obraz gry podzielony na kilka części. Centralne miejsce zajmuje uproszczony rysunek ulicy i domów oraz wędrującego ludzika – tropiciela historii. Wyżej przedstawiono obraz nieba. Na nim ukazują się symbole pożywienia – jabłka i dzbanki z napojem. Widoczny jest też kompas wskazujący zawsze północ. Jest nam bardzo pomocny w odnalezieniu wyjścia, które znajduje się na północnym zachodzie. Z

> 12

11 <

lewej strony u góry pokazuje się nazwa ulicy, po której idzie ludzik. Niżej objaśnienia związane z tą nazwą. Tylko jedno z nich jest prawidłowe. Gdy pojawi się na ekranie, należy potwierdzić jego prawidłowość przyciskając odpowiedni klawisz lub przycisk "fire" joysticka. Pole z prawej strony, to pole kontrolne gry. Umieszczono tu zarys planu miasta z poruszającą się kropką, która oznacza miejsce pobytu ludzika. Tu też są wskaźniki poziomu energii, czasu gry, liczby punktów oraz stoper uruchamiany przy odgadywaniu nazwy napotkanego pomnika lub bramy. Obok ukazują się klucze niezbędne do otwarcia zamkniętych bram.

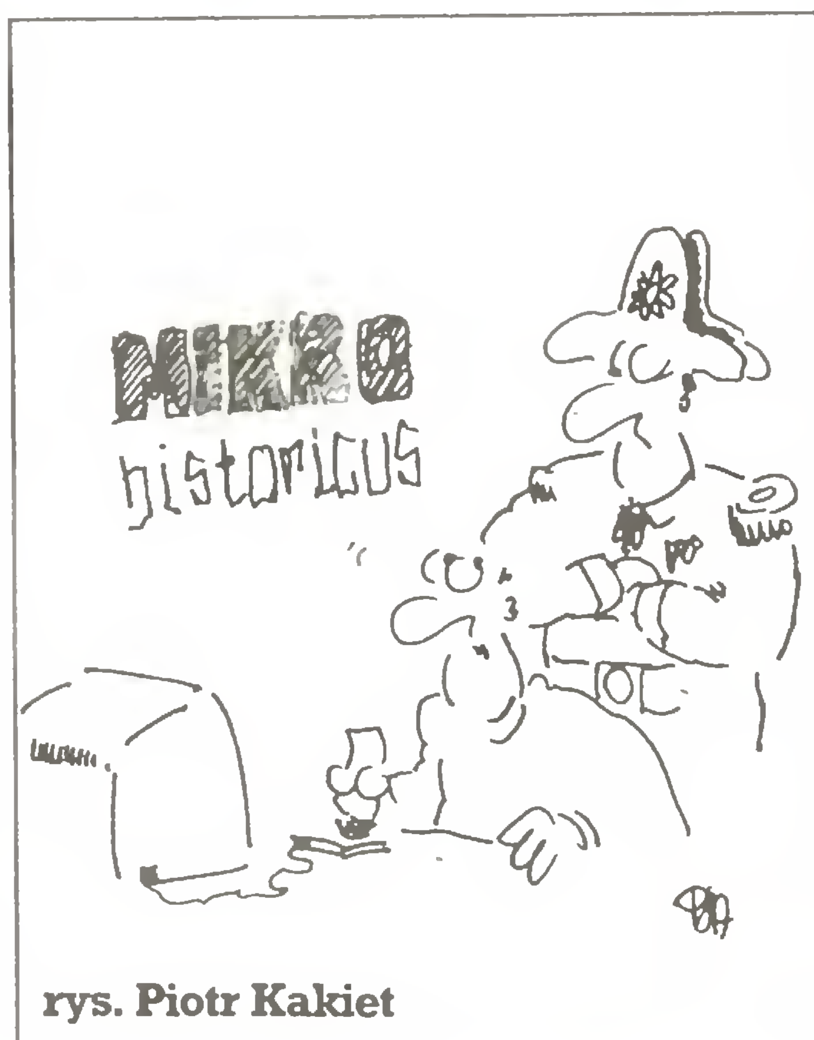
Ruszamy w drogę z południowo-wschodniego krańca miasta. Naszym celem jest jego przebycie do północno-zachodniej bramy w jak najkrótszym czasie i odpowiedzenie na jak największą liczbę pytań. Każda prawidłowa odpowiedź to dodatkowe punkty. Tylko wtedy, gdy zdobędziemy ich bardzo dużo, otrzymamy tytuł "supertropiciela historii". Jeżeli przekroczymy czas 50 minut, nie ukończymy gry i znajdziemy się w punkcie wyjścia. Także każda błędna odpowiedź powoduje cofnięcie o kilka ulic.

Idziemy ulicą, ukazuje się jej nazwa i dotyczące jej pytanie. Nazwa ulicy może być datą, wydarzeniem, nazwą organizacji, miejscowości lub nazwiskiem postaci historycznej. Na przykład "ul. 1466 roku", odpowiedzi: 1. Unia brzeska, 2. Koronacja Kazimierza Jagiellończyka, 3. II Pokój Toruński. Z tych odpowiedzi wybieramy prawidłową i możemy spokojnie wędrować tą ulicą dalej lub skręcić w napotkaną przecznicę. Jeśli nasza odpowiedź była błędna, spotka nas kara i przesuwamy się kilka ulic do tyłu. Czas na wybranie dobrej odpowiedzi, to czas przejścia ulicy. Jeśli ulica skończy się, a my nie udzielimy odpowiedzi, to drepczemy chwilę w miejscu, po czym przenosimy się na start. Możemy oczywiście, nie udzielając odpowiedzi, skręcić w inną ulicę, ale wtedy nie zdobędziemy dodatkowych punktów.

Udzielenie dobrej odpowiedzi może zostać nagrodzone małą niespodzianką – napotykamy pomnik ściśle związany z nazwą ulicy. Na przykład na placu Odsieczy Wiedeńskiej pomnik króla Jana III Sobieskiego. Jeśli chcemy odgadnąć przedstawioną na nim postać, podchodzimy jak najbliżej i naciskamy klawisz odpowiadający odpowiedzi "tak" (space lub "fire" joysticka). Ukazują się wtedy po kolei trzy nazwiska: Jan III Sobieski, Stefan Batory, Jan II Kazimierz. Przy każdym z nich musimy odpowiedzieć tak lub nie. Mamy na to 20 sekund. Czas jest odliczany na stoperze, który włącza się automatycznie, gdy wyrazimy wolę zgadywania nazwiska postaci z pomnika. Mamy trzy rodzaje pomników: postać w koronie i z berłem – władcy, dwie skrzyżowane szable na postumencie – żołnierze, postać stojąca – politycy.

Wędrując ulicami może się zdarzyć, że staniemy przed bramą. Na przykład na ulicy 1683 roku przed Bramą Wiedeńską. Nazwa bramy pochodzi zawsze od nazwy miejscowości związanej z nazwą ulicy. By przejść dalej, musimy ją otworzyć tzn. odgadnąć jej nazwę (tak jak przy pomniku) i mieć klucz – co jest dodatkowym utrudnieniem. W posiadanie klucza wchodzimy udzielając wcześniej dziesięciu prawidłowych odpowiedzi pod rząd. Niestety nie jest to łatwe. Przy bramie też obowiązuje limit 20 sekund. Jeśli czas upłynie, a my nie udzielimy odpowiedzi, to zarówno spod pomnika jak i spod bramy cofamy się na start. Niestety autorzy, by utrudnić grę, nie przewidzieli możliwości ruchu do tyłu. Pomnik możemy ominąć, bramy niestety nie. Dlatego odgadnięcie nazwy bramy przynosi największe korzyści. Otrzymujemy cztery dodatkowe punkty i dodatkową premię – przemieszczenie o kilka ulic do przodu.

Jak w każdym mieście tak i w tym zdarzają się ślepe uliczki. Musimy je rozpoznawać po nazwach nie związanych z historią. Natychmiast uciekamy z nich skręcając w najbliższą przecznicę. Gdy tego nie zrobimy, będziemy zaczynali grę od początku.



rys. Piotr Kakiet

Wędrówka po mieście zajmuje nam sporo czasu. Zdarza się, że zaskakuje nas noc. Miasto oświetlone jest wtedy tylko promieniami księżyca. Ciemna poświata obrazu powoduje utrudnienie w odczytywaniu nazw ulic. Nie wszystkie litery są widoczne. Prawdziwy tropiciel historii poradzi sobie jednak nawet z tym.

To wyludnione miasto nie jest tak spokojne na jakie wygląda. Grasują w nim różne duchy, dobre i złe. Pojawiają się nagle na niebie i spadają na ludzika. Jeśli trafimy na złego ducha, zaczniemy grę od początku lub w najlepszym razie cofniemy się o kilka ulic. Dobre duchy znikają nie wyrządzając krzywdy. Spotkanie z duchami nie należy do przyjemnych i trzeba ich unikać.

Długa wędrówka męczy i wyczerpuje siły. Od czasu do czasu pojawia się pożywienie. Spadają jabłka i dzbanki z napojami. Musimy tak poruszać ludzikiem, by spadały na niego. "Zjedzenie" jabłka lub "wypicie" napoju przywraca maksimum energii. Choć nie zawsze. Trzeba uważać na napoje. Dzbanki zawierają wodę lub wino. Jeśli będzie to dzbanek z winem, to natychmiast odczujemy zgubny wpływ alkoholu. Ludzik nie będzie dokładnie reagował na nasze polecenia, krok jego stanie się chwiejny, a głowa będzie się kiwać na boki. Nie potrafi też przyspieszyć kroku.

Gdy pokonamy wszystkie niebezpieczeństwa i dotrzemy do północno-zachodniej bramy, okaże się czy zasługujemy na miano "supertropiciela historii". Na sumę zdobytych punktów składają się punkty za dobre odpowiedzi i premia za czas przebycia miasta np. za 30 minut 20 punktów. Tylko suma powyżej 150 punktów zapewnia ten tytuł. Nie jest łatwo ją osiągnąć, ale próbować trzeba.

Program "Miasto Historii" jest programem interesującym. Uczy szybkiego kojarzenia faktów historycznych z odpowiednimi datami, postaciami historycznymi i ich rolą, podaje podstawowe informacje. Dotyczy to historii Polski, od Mieszka I po Bolesława Bieruta. Biorąc pod uwagę objętość materiału i pamięci komputera, nie należy się dziwić, że są uwzględnione tylko najważniejsze wydarzenia. Forma, w jakiej zrealizowano ten test – grę jest efektowna. Prosta grafika, ruch na ekranie (namiastka animacji), możliwość wyboru działań przez gracza (niestety zbyt ograniczona – brak możliwości cofania ludzika) to zalety programu. Wady to przede wszystkim bardzo częste odsyłanie gracza do miejsca startu. O ukończeniu gry z sukcesem decyduje często tylko jedno pytanie. Błędna odpowiedź na pytanie o nazwę bramy eliminuje gracza nawet, gdy jest to ostatnia brama przed metą. Zdobyć klucz też niełatwo. To może być zniechęcające szczególnie dla uczniów, którzy mają mniejszy zasób wiadomości i program służy im nie tylko do sprawdzenia swych umiejętności, ale do nauki. Program utrwala również stereotyp patrzenia na naukę historii, jako wkuwanie dat, miejscowości, bitew i nazwisk.

Uwagi te nie umniejszają faktu, że jest to jedna z najlepszych realizacji testu, jaką znam. W łatwy sposób można zmienić zawartość merytoryczną programu tworząc na przykład miasto chemii czy biologii. To też duża zaleta. Nie jest to jednak program, który można było by wykorzystać na lekcji w klasie. Na pewno przyda się w kółku historycznym czy w domu.

Kurier

Michał Setlak

Drugi ruch głową

Na początku maja w Dobieszkowie koło Łodzi odbyło się Drugie Międzynarodowe Sympozjum Studenckich Kół Naukowych na temat "Mikrokomputery w nauce i technice".

Opowieść

Słońce kończyło już swą wędrówkę po bezchmurnym niebie, gdy na skraju wioski zamajaczyły przede mną zabudowania Centralnego Ośrodka Szkoleniowego OHP. Gości przywiózł tu wcześniej autokar. Ja – wstyd się przyznać – spóźniłem się na uroczyste otwarcie. Do Dobieszkowa dotarłem per pedes (i per PKS) w porze ogniska, które wieńczyło program inauguracyjnego dnia imprezy.

Stara to prawda, że nic tak nie brata ludzi, jak miejsce we wspólnym kręgu i śpiew przy akompaniamencie gitary. Było nas około pięćdziesiątki; śpiewali wszyscy – Algierczycy, Czesi, Francuzi, Kanadyjczycy, Polacy i Ukraińcy. Rozchodziliśmy się nie pamiętając już o dzielących nas barierach. Rychło też przekonałem się, że nic nie nastraja lepiej do pracy niż wspaniałe warunki pobytu, jakie zapewnili studentom organizatorzy.

Ad rem...

Wachlarz tematów prezentowanych w Dobieszkowie przez studencką brać był bardzo szeroki. Obejmował zarówno problemy popularnego oprogramowania i wykorzystania mikrokomputerów w dydaktyce, jak i w pełni profesjonalne zastosowania w zaawansowanej technologii.

Kanadyjczycy mówili o algorytmach kodowania danych w systemach łączności satelitarnej – i o programach *Public Domain*. Nawiasem mówiąc, niespodzianką była dla nich informacja o działalności sieci Fido w Polsce.

Czesi zaprezentowali zgrabny syntezator mowy zaprojektowany jako urządzenie zewnętrzne mikrokomputera, uświadamiając słuchaczom, że nie tak łatwo nauczyć komputer mówić po czesku. I nie sposób go słuchać nie uśmiechając

się. Przedstawili także swoje prace w dziedzinie hiperstabilnych układów automatyki.

Również automatyki dotyczyły referaty Francuzów; pierwszy z nich opisywał system diagnostyczny silnika helikoptera, drugi, którego autorką była zresztą Polka na stałe mieszkająca we Francji, w bardzo ciekawy sposób objaśniał termograficzną metodę testowania jakości montażu powierzchniowego obwodów elektronicznych. Trzeci z Francuzów opowiedział o gromadzeniu danych na nośnikach optycznych.

Gość z Kijowa, profesor Walery Owsianik, przybliżył teoretyczne i praktyczne aspekty automatyzacji pomiarów.

Algierczycy podzielili się z audytorium swymi doświadczeniami w zakresie diagnozowania wibracji i ich symulacji numerycznej.

W referatach gości zagranicznych komputery wymieniane były tylko z nazwy, zaś opisy oprogramowania wspierane były jedynie przeźroczami. Większość Polaków demonstrowała swoje programy "na żywo". Nie mogliśmy się, co prawda, pochwalić stacjami Sun, o których mówili Kanadyjczycy, był jednakże jeden IBM PC/AT 386, Amstrad CPC 6128 i kilka "bardzo mikrokomputerów" Spectrum, co w zupełności wystarczyło. Nawet chwilowe przerwy w dopływie prądu nie złamały ducha walki i nasi wypadli całkiem nieźle (ograniczenie czyni mistrza!).

Trio Łodzian pokazało "System testowania nabytej wiedzy", który zmienia poczciwego "peceta" w postrach studentów. "Ośmiobitowce" znakomicie sterowały eksperymentem naukowym i obliczały parametry ruchu pojazdów. Okazało się, że nawet skromne możliwości obliczeniowe i graficzne Spectrum pozwalają symulować procesy produkcyjne lub projektować części maszyn. Bardzo ciekawe okazało się połączenie grafiki Spectrum (wykorzystano program Jacka Potempy "Grafika") z techniką wideo; powstałe w ten sposób filmy mogą znacznie urozmaicić wykłady i usprawnić proces dydaktyczny.

Tak ogromna ilość informacji mogłaby przyprawić o niestrawność nawet słonia. Na szczęście gospodarze zadbali o zdrowie uczestników. Referaty serwowano po dwa – trzy w przerwach między posiłkami, zaś na przekąskę oferowano dwa spektakle w łódzkim Teatrze Wielkim: specialite de la maison - "Straszny Dwór" à la Moniuszko, oraz "Cyrulika Sewilskiego". Miłośnicy lżejszej strawy duchowej woleli wieczorne spotkania z gitarą. Polecam ten przepis – właściwe proporcje pracy i wypoczynku.

Sprawcy i motywy

Nie od dziś wiadomo, że student to stworzenie leniwe, lecz posiadające ogromne zasoby energii, zdolne nawet do kruszenia murów. Jak jednak wyzwolić tę energię, skierować we właściwym kierunku, zaprząc do pożytecznej pracy?

Receptę na to znaleźli pomysłodawcy i organizatorzy Sympozjum: dziekan Wydziału Mechanicznego Politechniki Łódzkiej, prof. dr hab. Jan Krysiński, dr Stanisław Mitura i dr Tadeusz Pacyniak wspólnie z Kołem Naukowym Inżynierii Materiałowej, Mikrokomputerowym Kołem Naukowym Mechaników, Kołem Naukowym Energetyków, Kołem Naukowym Odlewników, Studenckim Kołem Naukowym Sieci Komputerowych. Pokonując liczne przeszkody, znajdując odpowiednich sponsorów (dziekan Wydziału Mechanicznego P.Ł., rektor P.Ł. ds. studenckich, Sp. z o.o. Inter Global), wykorzystując własne kontakty, zorganizowali atrakcyjną imprezę. Wielu studentów z chęcią wzięłoby udział w takim sympozjum, aby jednak zakwalifikować się do grona uczestników, należało przedstawić własną pracę. To najlepszy bodziec do twórczego działania! Niebagatelne znaczenie ma także udział gości zagranicznych. Po pierwsze, podnosi rangę przedsięwzięcia, po drugie... w przyszłym roku najlepsi pojadą w drodze wymiany do Francji, Kanady... Czyż to nie kusząca perspektywa? Wystarczy tylko ruszyć głową.

\Kurier\

Jacek Szelożyński

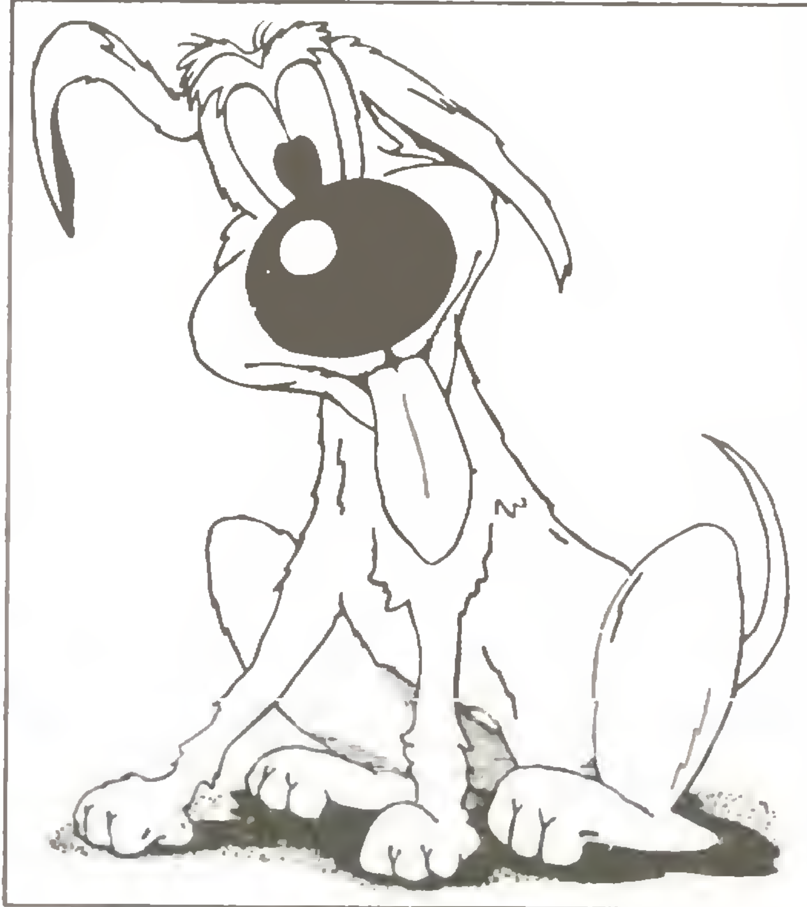
Od Wildcata do Quick Cata

Od czasu napisania artykułu o kocie ("Dziki kot z Wybrzeża" – "Komputer" 2/89) nasza sieć zdążyła przekształcić się w szybkiego kota!

Quick CAT to nowa sieć oparta na programie Quick BBS, który zasadniczo oferuje to samo, co Wildcat! lub Fido, jednakże robi to szybciej i lepiej.

Co wyróżnia Quick BBS spośród innych programów?

Przede wszystkim zgodność formatu poczty z normami IFNA, czyli Międzynarodowego Stowarzyszenia Sieci Fido. Stawiam tę cechę zdecydowanie na pierwszym miejscu, ponieważ bez takiej zgodności żaden program BBS nie ma szans na włączenie się do światowej sieci Net– i Echomail. Po pierwszych doświadczeniach i "zabawach" z siecią każdy użytkownik chciałby nieco więcej niż tylko pobierać i posyłać pliki lub pocztę lokalną. Powoli wciąga go idea szerokiej wymiany informacji (prawdopodobnie w chwili, gdy będziecie czytać te słowa Quick Cat będzie już aktywnym, tj. ogólnodostępnym punktem sieci Ainex RBBS z Holandii). Nasze działy pocztowe oferują wymianę doświadczeń na temat klonów IBM PC oraz komunikacji modemowej (stąd wysyłamy i otrzymujemy listy do i z całej Europy Zachodniej) oraz pogawędki hobbystów z różnych dziedzin (cały świat!).



Nie zapomnieliśmy o miłośnikach wymiany (czy raczej pobierania...) nowych programów. Z myślą o nich Quick Cat oferuje ponad 400 różnych plików, które można przesłać za pomocą najlepszego protokołu transmisji – Zmodem. Jeśli z całego zestawu plików w formacie ARC zainteresowani jesteśmy tylko jednym, można najpierw obejrzeć zawartość spakowanego pliku, po czym korzystając z opcji ARC to ARC pobrać wybrany program. Tej możliwości nie oferuje chyba żadna inna sieć poza Quick BBS.

Dla niecierpliwych (Quick Cat!) jeszcze jedno udogodnienie – nie trzeba czekać na ukazanie się całego menu. Na przykład aby dotrzeć do 6 działu plików i obejrzeć listę dostępnych programów naciska się kombinację klawiszy F-6-L, trzeba po prostu pamiętać potrzebny ciąg klawiszy. Na początku sesji nie należy czekać na ukazanie się całego ekranu powitalnego – wystarczy naciśnięcie S i ukaże się główne menu!

Quick BBS daje możliwość pracy z programem "na żywo" (*on- line*, podczas sesji). W tym celu zamierzamy wykorzystać jakiś ciekawy program, który pozwoli przeglądać i przeszukiwać rekordy dużej bazy danych. Na razie programem *on-line* jest zbiór ciekawostek o różnych sławnych ludziach i ważnych wydarzeniach, pokazujący codziennie inny zestaw wiadomości.

To są najistotniejsze zmiany, jakie wprowadza Quick Cat w porównaniu z siecią Wildcat! Zapewne Czytelnik chciałby wiedzieć, czy nowy "kotek" rozumie po polsku? Jak najbardziej, jest nawet dwujęzyczny! Jeśli wybierze się pracę z atrybutami ANSI, wówczas wszystkie menu i większość komunikatów ukażą się po polsku. Gdy bardziej odpowiada angielski – proszę bardzo, dwa naciśnięcia klawiszy i już pracujemy bez ANSI co prawda, ale za to w języku angielskim.

W nowej sieci celowo zrezygnowaliśmy z pełnego spolszczenia – jesteśmy teraz węzłem międzynarodowym i użytkownik zagraniczny powinien mieć również możliwość pracy z naszym BBS-em. Obserwując sieci holenderskie lub belgijskie nie zauważyłem ani jednej, która pracowałaby wyłącznie w ojczystym języku tamtejszego *Sysopa*, co więcej komunikaty angielskie stanowią 90% wszystkich ukazującycych się tekstów. Znam tylko jedną sieć, która oferuje pracę dwujezyczną – jest to płatna sieć profesjonalna NEA BBS.

Przy okazji chciałbym poruszyć temat, który czasem przewija się w listach i rozmowach telefonicznych z naszymi użytkownikami. Chodzi o samodzielne założenie sieci. Doskonałym komentarzem jest notatka autorstwa Stevena K. Hoskina (1:128/31) zamieszczona w biuletynie *Fido News* z dnia 27 lutego 1989 roku:

Epizod 22 – Po co ludzie uruchamiają sieci?

Tak, mamy bardzo dziwne hobby. Wydajemy dużo pieniędzy na komputer, potem jeszcze więcej by dokupić do niego twardy dysk. Do tego dochodzi wydatek na zakup modemu. Z kolei odkrywamy jakiś ciekawy program i decydujemy, iż właśnie na nim uruchomimy naszą sieć. W niektórych przypadkach wiąże się to z dokupieniem większego dysku twardego i, być może, szybszego modemu. Walczymy z nieprzyjaznym oprogramowaniem i ubogą dokumentacją. Przebijamy się z trudem przez programy pomocnicze mogące nadać naszej sieci wymagany kształt i zmieniamy oprogramowanie, gdy tylko pojawia się coś nowego, a posiada jakąś cechę, która się nam spodobała. Spędzamy całe godziny na konfigurowaniu programu i plików pomocniczych. Czekamy z niecierpliwością na automagiczne wykonanie pewnych zaprogramowanych wydarzeń w sieci, które ze względu na koszty połączeń mają zwykle miejsce późną nocą. Chuchamy, dmuchamy i lamentujemy nad naszymi maszynami oraz mozolnie dłubiemy w nowych konfiguracjach. Dochodzimy do tego, jak robi się pewne rzeczy i narzekamy, że inne nie chcą działać. I wszystko po to, aby umożliwić pracę z naszym komputerem INNYM ludziom!

Chwileczkę... Gdzie tu logika? Cóż, większość "ciężko zakocha-

> 14

13 <

nych" powie, że to dla samej radości prowadzenia sieci. Niektórym satysfakcję daje to, że pomagają innym ludziom (t.j. użytkownikom). Jeszcze innych cieszy mistrzowska precyzja systemu i oprogramowania. Są tacy, którzy podejmują wyzwanie ze strony techno-magii, uprawiając sztukę dla sztuki. A dla niektórych jest to prosta ciekawa idea.

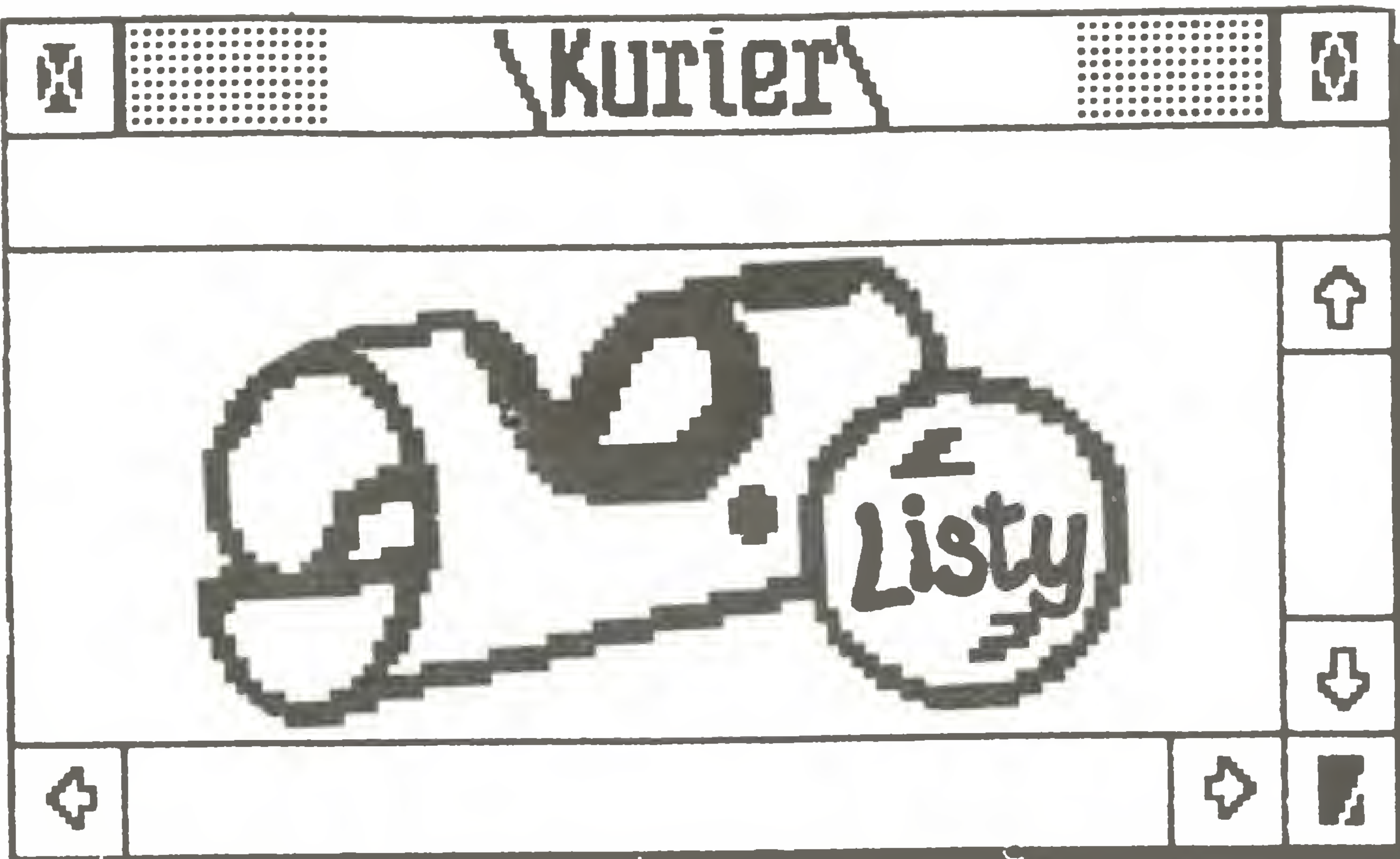
Jak by na to nie patrzeć to jest hobby. Niektórzy traktują je bardziej, inni mniej poważnie, ale z pewnością jest to coś, bez czego w sumie można by się obejść. Jest to sposób na wypełnienie wolnego czasu. Zgoda, nawet CAŁEGO wolnego czasu, i to wszystko. Wszelkie korzyści – pomoc, mistrzowska precyzja, wyzwanie, nostalgia to tylko dodatki do całego BBS-u.

Niestety, jak zawsze w życiu, tak i tutaj korzyściom towarzyszą pewne niewygody: późne godziny, frustrujący proces konfigurowania systemu, zawód, złość, wydatki na sprzęt, próby włamań ze strony włamywaczy i wreszcie niespodziewany atak wirusów. SAM musisz zdecydować, czy to wszystko warte jest twoich wysiłków. Czy plusy przeważają nad minusami? Jeśli w głębi duszy uznasz, że nie – nie zakładaj sieci. Działaj jako punkt wymiany poczty lub ograniczn się do roli użytkownika. Ale nie kieruj do mnie zażaleń z powodu złego wyboru. TY SAM go dokonałeś.

Ps. Od Sysopa Quick Cata: Instalowanie nowej sieci Quick BBS zajęło mi ponad miesiąc (od 5 lutego do 10 marca). W tym czasie z własnej nieprzymuszonej i fanatycznej woli przesiadywałem nad tym od 9 rano do 17 i od 18.30 do północy. Dzień w dzień. To jest naprawdę olbrzymia radość i satysfakcja zobaczyć, że coś działa dokładnie tak, jak sobie wymyśliłem!

Po otrzymaniu drugiej linii telefonicznej przystąpimy do działania w stronę rozwiązań profesjonalnych. Będzie to najprawdopodobniej wielka i płatna baza lub bazy danych, być może połączone z usługami telexowymi i faxowymi. BBS pozostanie wówczas, w dotychczasowych godzinach właśnie, jako czysta radość tworzenia i zabawy nowoczesnym sprzętem. Jednak od czegoś trzeba zacząć – pierwsze próby z modemem i założenie sieci pozwoliły zdobyć olbrzymią ilość nowej wiedzy, bez której nie moglibyśmy nawet marzyć o profesjonalnym podejściu do łączności modemowej.

Zapraszamy wszystkich obecnych i nowych użytkowników sieci do dalszej współpracy: Gdańsk (058) 523319, codziennie od godz. 22.00 do 8.55.



Jest wiele głosów donośnych jak trąby, popierających potrzebę szybkiego rozwoju i unowocześnienia gospodarki. I cóż się dzieje, znienacka wbija się tej idei nóż w plecy. Wprowadzona 136-krotna podwyżka opłat za korzystanie z modemów i wymianę informacji woła o pomstę do nieba. Nie można tego wytłumaczyć tzw. uruchamianiem kosztów. To jest po prostu "czyszczenie" portfeli. Nie wspomnę o sprawie jakości polskich łączy telefonicznych, która może się wiązać np. z obniżeniem prędkości transmisji (co za tym idzie dodatkowym zarobkiem przedsiębiorstwa PPTiT).

Moje pytanie – propozycja jest następująca: Czy nie można by stworzyć organizacji chroniącej prywatnych użytkowników mikrokomputerów przed takimi opłatami. Czy jakaś organizacja, choćby PTI, nie mogłaby wystąpić z postulatem obniżenia opłat modemowych dla indywidualnych użytkowników?

Kolejna prośba dotyczy opublikowania informacji o stanie prac nad prawną ochroną oprogramowania w Polsce. Po kilku konferencjach i wypowiedziach ucichło. Wasza rozmowa z p. Kajkowskim jeszcze raz pokazuje problemy ludzi, którzy chcą się zajmować tą dziedziną. Swego czasu snułem plany stworzenia firmy zajmującej się produkcją i sprzedażą oprogramowania. Doszedłem jednak do wniosku, że nie ma szans na ekonomiczną wydajność takiej inwestycji bez wspierania jej np. handlem komputerami (pośrednictwem).

Piotr Grabowski
Elbląg

* * *

Chciałbym dorzucić swoje "trzy grosze" do dyskusji o wirusach. Jeżeli ktoś pracuje głównie ze stacją dyskietek, to dobrą praktyką jest przechowywanie programów (z rozszerzeniem .COM i .EXE) na jednej dyskietce, zabezpieczonej przed zapisem danych przez zaklejenie bocznego wcięcia (przy okazji zadanie dla Terminatora Terminologicznego: jak przetłumaczyć na język polski słowa *"notch"* i *"protect tab"*?). Chociaż metoda ta wprowadza konieczność przechowywania danych na oddzielnym dysku, zapewne w większości przypadków uchroni użytkownika przed niepożądaną infekcją. Nie jest to jednak pewność stuprocentowa, gdyż programy nie używające mechanizmów DOS mogą dokonywać zapisu nawet na dyskietkach z zaklejonym wcięciem. Co gorsza, czytałem gdzieś jakoby niektóre wirusy były w stanie uszkodzić stacje dyskietek w sensie fizycznym przez bezpośrednie adresowanie sterownika. Z nadzieją, że nikt z nas nie nabawi się takiej "grypy", przesyłam pozdrowienia dla redakcji i wszystkich Czytelników.

Krzysztof Włodarski
Flemington, USA

Od redakcji: *zapis na dyskietkach we wszystkich znanych nam stacjach dysków jest niemożliwy. Chyba że wcześniej sprzętowo "odbezpieczono" stacje odłączając stosowny kabelek.*

* * *

Dałbym dużo, żeby dowiedzieć się, kto, personalnie, wpadł na tak poroniony pomysł, aby w STraganie zamieścić na pełnych dwóch stronach materiał Stefana Nawrockiego "Polskie znaki w Word Plusie?!"

Człowiek używający od roku Word Plusa w wersji 2.02, wyposażonej w polskie znaki na ekran i drukarkę oraz w parę wersji konfiguracji klawiatury, dowiaduje się, jak może sobie zdefiniować i zainstalować polskie znaki do tegoż edytora! (notabene wcześniejsza znana mi wersja tegoż edytora - 1.24 - również posiada zestawy polskich liter na ekran i drukarkę).

Jeśli już chce się uhonorować p. Nawrockiego, niewątpliwie wybitnego programistę, to przecież można zwrócić się do niego, żeby napisał np. driver dla drukarki Star LC-10, której Word Plusowi (i nie tylko) brak, a która jest coraz popularniejsza na naszym rynku. Albo pożyteczne urządzenie, pozwalające wybierać z pulpitu roboczą stację, bez wiadomych kombinacji. Tego też bardzo brak Word Plusowi. Można też poprawić Tempus tak, by nie zawieszał się przy nowym TOS-ie. I tę poprawkę warto opublikować.

Dajmy już sobie w ogóle spokój z tymi polskimi znakami na ekranie. Są już programiki pomocnicze, pozwalające przyporządkować odpowiednie znaki (już zdefiniowane) wybranym klawiszom itd., itp.

I na koniec głos (łagodnie) krytyczny:

Duża część waszych materiałów powstaje na edytorach tekstów z wykorzystaniem funkcji dzielenia słów. Więc zastrzeżcie, żeby autorzy tę funkcję wyłączali, albo niech wasza korekta przykłada się do pracy i koryguje horrendalne błędy, jak przeniesienia w obrębie rdzenia wyrazu. Bo od tego aż coś ściska w środku.

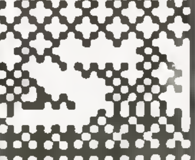
Życzę satysfakcji z pracy i soczystszych jej owoców dla nas.

Witold Radwański
Warszawa

Od redakcji: Z przykrością odnotowujemy przykład pokutującej w komputerowym światku niefrasobliwości. Tym większe rozczarowanie, że Czytelnik kończy list szczerymi życzeniami owocnej pracy. Cóż to bowiem znaczy "poroniony pomysł" w odniesieniu do bardzo wartościowej pracy wykonanej dla powszechnego pożytku, niosącej bezdyskusyjne wartości inspirujące i szkoleniowe? Co należy rozumieć przez "człowiek używający... wyposażonej w polskie znaki..."? Skąd Szanowny Czytelnik ma polskiego word Plusa, skąd ma oprogramowanie "download" do drukarki? Co to znaczy "wcześniejsza wersja również posiada..."? Czyżby spadło z nieba, a może dotarło drogą hipnozy? Bo jeśli Czytelnik sam napisał programy i dokonał niezbędnych restylizacji, nigdy nie wpadłby na pomysł nieszanowania pracy wybitnego polskiego programisty.

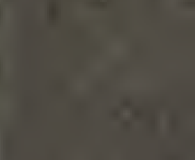
Co wreszcie znaczy zestaw sugestii pod adresem p. Nawrockiego, jeśli idzie o jego plany i ambicje programistyczne? Pan Nawrocki i jego zasługi nie wymagają specjalnego "honorowania". Jaka też myśl kierowała ręką Czytelnika, gdy proponował driver na LC-10, drive selector i poprawki do "Tempusa"? Proszę Czytelnika, to jest praca, za którą trzeba zapłacić godziwe pieniądze. To nie jest działalność klubowa.

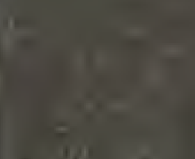
Na koniec istotna uwaga. Kto się zna na przedmiocie, wie, że błędne przenoszenie słów nie ma nic wspólnego z tą opcją obecną i aktywną na etapie pisania tekstu. Dzielenia dokonuje fotoskład na podstawie mnóstwa informacji o stopniu, kroju i szerokości składu determinowanego polskimi normami poprawnościowymi. Mylne jest zatem mniemanie Czytelnika, że autor tekstu ma wpływ na tę fazę produkcji numeru. Niewielki wpływ na ten etap produkcji ma również redakcja.



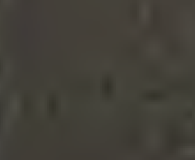












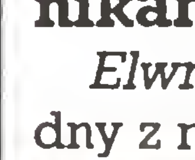














\W pracy\

Witold R. Rudolf

Z taśmy na dyskietkę

Polski mikrokomputer edukacyjny Elwro 800 Junior znalazł się już w wielu szkołach. Niestety zasób wiadomości o tym komputerze jest ograniczony, dlatego też chciałbym się podzielić swoimi doświadczeniami z jego użytkownikami.

Elwro 800 Junior jest w pełni zgodny z mikrokomputerem ZX Spectrum, który jest obecnie najbardziej rozpowszechniony w naszych szkołach i w związku z tym często wraz z Juniorem stanowi wyposażenie laboratoriów szkolnych. Oprogramowanie rezydujące Juniora, zgodne z oprogramowaniem ZX Spectrum, zostało przy tym wzbogacone, w szczególności o operacje dyskowe i sieciowe.

Zaletą ZX Spectrum jest bardzo bogate oprogramowanie. Wiele z tych programów znajduje się już w szkolnych zbiorach, choć na bardzo niewygodnym nośniku, jakim jest taśma.

Ponieważ Junior może wykonywać programy napisane dla ZX Spectrum, spróbowałem opracować prostą metodę przenoszenia programów z taśmy na dyskietkę i rozsyłania ich w sieci lokalnej.

Na początek dwie uwagi ogólne.

Ze względu na dużą zawodność stacji dysków (korzystam z FD-600-2) radzę przechowywać programy na co najmniej dwóch dyskietkach – często zdarza się, że stacja rozformatowuje dysk, a wtedy nie uda się już odzyskać programu.

Nie wszystkie programy z ZX Spectrum działają poprawnie w Juniorze, np.: gra Jet Pac. Może ktoś posiada wersję działającą na Juniorze, będę wdzięczny za informację; Lmowa - nie działa z dyskietki (?); Survival, Podział Godzin – programy te mają fragment w kodzie maszynowym, który automatycznie się uruchamia.

Programy jednoblokowe

Jeśli program uruchamia się automatycznie, to za pomocą programu COPY COPY sprawdzamy, od którego wiersza uruchamiany jest przenoszony program (pole START).

Wgrywamy program do komputera (LOAD " "). Jeśli uruchamia się automatycznie, przerywamy jego wykonywanie (SHIFT+SPACJA lub gdy to nie skutkuje, podajemy błędne dane np. literę zamiast liczby).

Nagrywamy program na dyskietkę: **SAVE *"nazwa"** lub, gdy program ma się uruchamiać od wiersza START: **SAVE *"nazwa" LINE start.**

Program zostanie zapisany na dyskietce w pliku NAZWA.PRG - wczytać go można przez **LOAD * "nazwa".**

Nazwę programu należy tak dobrać, aby odpowiadała jego treści.

Programy wieloblokowe

Proponuję przyjąć następujący standard dla nazw bloków na dysku: Nazwa_Podstawowa – 7 znaków, kolejne bloki otrzymują naz-

> 16

15 <

wę Nazwa_PodstawowaNr_bloku, np. trzyblokowy program VU-CALC:

VU-CALC
VU-CALC1
VU-CALC2

Każdy blok sprawdzamy za pomocą programu COPY COPY i notujemy:

- dla bloku typu P (program) numer wiersza, od którego program jest uruchamiany (pole PROG) – na ogół programy wieloblokowe startują automatycznie;
- dla bloku typu B (bajty) początek (pole START) i długość bloku (pole DLUG).

Uwaga: blok o danych: START=16384 i DLUG= ok. 6900 odpowiada blokowi SCREEN$ a nie CODE, można to dodatkowo sprawdzić uruchamiając program.

Przy blokach typu P (program) postępujemy następująco:

- wczytujemy blok do komputera (LOAD" ") i przerywamy jego pracę (SHIFT+SPACJA),
- przeglądamy program zmieniając instrukcje LOAD i SAVE: LOAD "nazwa" na LOAD *"Nazwa_Dyskowa" LOAD "nazwa" SCREEN$ na LOAD *"Nazwa_Dyskowa" SCREEN$ LOAD "nazwa" CODE na LOAD *"Nazwa_Dyskowa" CODE jeśli a$ jest zmienną, to LOAD a$ CODE na LOAD *a$ CODE

Podobnie postępujemy dla instrukcji SAVE.

Uwaga: jeśli nie możemy przejrzeć programu, to na ogół pomaga wykonanie INK 0 lub INK 7 (zmiana koloru tekstu) lub, gdy wyświetlany jest wiersz zawierający REM i ciąg znaków, LIST od numeru wiersza większego niż widoczny.

- nagrywamy program na dysk podając odpowiednią nazwę i numer wiersza początkowego (jak wyżej).

Bloki typu SCREEN$ przenosi następujący program:

```
10 LOAD " " SCREEN$
20 SAVE *"Nazwa_Dyskowa" SCREEN$
```

Bloki typu CODE przenosi program:

```
10 LOAD " " CODE start, dlug
20 SAVE *"Nazwa_Dyskowa" CODE start, dlug
```

gdzie start i dlug to dane odczytane za pomocą programu COPY COPY.

Rozsyłanie programów w sieci

Oto moja propozycja rozsyłania programów w sieci. Na początku dobrze jest zapisać na dyskietce pod nazwą "SIEĆ" następujący programik:

```
9990 INK 7: PAPER 0: CLS: PRINT "PODAJ ADRES SIECIOWY:"
     : PRINT "0 – KONIEC, 1 – URUCHOMIENIE"
9991 INPUT ADRES: CLS: IF ADRES=0 THEN STOP
9992 IF ADRES=1 THEN RETURN
9993 SAVE@ ADRES,"NAZWA" LINE 9997: PAUSE 200
9994 SAVE@ ADRES,"NAZWA" CODE start, długość: PAUSE 200
9996 GOTO 9990
9997 REM Tu wpisać odpowiednie dane początkowe
9998 PRINT "POCZEKAJ NA ZNAK NAUCZYCIELA": PAUSE 0
9999 GOTO adres_startu
```

Dla uproszczenia założyłem, że wszystkie programy kończą się na wierszu 9989. W innym przypadku trzeba będzie do tego doprowadzić lub zmienić numerację wierszy.

Program ten pozwala rozsyłać w sieci programy jedno- i wieloblokowe, tak do wszystkich komputerów (adres 255), jak i do ich grup. Pauzy w wierszach 9993 i 9994 są konieczne ze względu na dość wolną transmisję w sieci. Podobnie wiersz 9998 – służy do wstrzymania biegu programu do chwili zakończenia transmisji wszystkich bloków (sygnałem tego będzie ponowne zgłoszenie się programu rozsyłającego w komputerze nauczycielskim). Wtedy uczeń powinien nacisnąć dowolny klawisz w celu uruchomienia programu. W wierszu 9997 należy umieścić wszelkie instrukcje PAPER, INK, BORDER i CLEAR wykonywane w programie wieloblokowym przed wczytaniem kodu.

Z rozsyłania "okładek" (ekranów – SCREEN$) rezygnujemy.

Do programu jednoblokowego dołączamy program SIEĆ przez **MERGE *"SIEĆ"**. W wierszu 9992 zamiast RETURN wstawiamy GOTO adres_startu, (adres_startu to wiersz, od którego program ma rozpocząć działanie). Podobnie wstawiamy adres_startu w wierszu 9999. Usuwamy wiersz 9994 i zapisujemy gotowy program na dyskietce za pomocą instrukcji **SAVE *"Nazwa_dyskowa" LINE 9990.**

W programie wieloblokowym musimy odszukać część programu w Basicu, która ładuje ostatni blok w kodzie. Dołączamy do niej program SIEĆ i bezpośrednio po ostatniej wykonywanej instrukcji LOAD wstawiamy GOSUB 9990. Teraz w wierszu 9994 i ewentualnie 9995 podajemy dane o wszystkich fragmentach kodu, które były ładowane (w całym programie). Start i długość tych fragmentów możemy odczytać w podany już sposób. W wierszu 9997 trzeba natomiast wstawić wszystkie instrukcje PAPER, INK, BORDER i CLEAR wykonywane w programie przed wczytaniem kodu. W wierszu 9999 podajemy numer wiersza, od którego startuje przerabiana przez nas część.

Po tych przeróbkach zapisujemy tę część na dysk za pomocą rozkazu **SAVE *"Nazwa_dyskowa" LINE** start.

Uwaga: jeżeli program składa się z kilku kolejno wykonywanych w Basicu fragmentów, przeróbka na wersję sieciową może się nie udać.

Obsługa zestawu programów na dysku

Załóżmy, że na dyskietce znajdują się trzy programy o nazwach PIERWSZY, DRUGI i TRZECI. Nie jest istotne, czy są one jedno- czy wieloblokowe – ważne tylko, że wykonanie np. instrukcji **LOAD * "DRUGI"** spowoduje uruchomienie programu DRUGI. Proponuję obsługiwać dysk za pomocą programu MENU:

```
10 INK 7: PAPER 0: CLS: PRINT "1 – PIERWSZY"
20 PRINT "2 – DRUGI"
30 PRINT "3 – TRZECI"
40 REM Tu można dopisać informację o nowych programach
1000 INPUT NUMER: GOTO 1000+10*NUMER
1010 LOAD *"PIERWSZY"
1020 LOAD *"DRUGI"
1030 LOAD *"TRZECI"
1040 REM Tu można dodać ładowanie nowych programów
```

Pozwoli to uniknąć kłopotów związanych z dużą ilością plików na dyskietce zawierającej programy w Basic Spectrum.

Na koniec zastosujmy zdobytą wiedzę. Dla przykładu podaję, jak przenieść na dyskietkę szeroko rozpowszechniony program kalkulacyjny VU-CALC. Zbudowany jest on z trzech bloków: głównego (typu P), "okładki" (SREEN$) i kodu maszynowego (typ B).

Wczytujemy kolejne bloki za pomocą programu COPY COPY. Notujemy dane:

- główny blok – typ P (w Basicu), start w wierszu 10,
- drugi blok – SCREEN$,
- trzeci blok – typ B (bajty), start =25232, długość=5270.

Cofamy taśmę i wczytujemy pierwszy blok przez **LOAD " "**. Po przerwaniu (SHIFT+SPACJA) na ekranie nie ma tekstu programu. Musimy zmienić kolor liter (INK 7). Przeglądamy program i zmieniamy:

- w wierszu 1000 LOAD a$CODE na LOAD *a$CODE,
- w wierszu 4000 SAVE a$CODE na SAVE *a$CODE.

Teraz dołączamy program SIEĆ (**MERGE *"SIEĆ"**) i zmieniamy cztery wiersze:

```
9993 SAVE @ ADRES,"VU-CALC" LINE 9997: PAUSE 200
9994 SAVE @ ADRES,"VU-CALC1" CODE 25232, 5270: PAUSE 200
9997 CLEAR VAL "25231"
```

(BORDER, PAPER i INK nie musimy ustawiać – maskowały one tylko ładowanie kodu)

```
9999 GOTO 3200
```

oraz w wierszu 10 cbie instrukcje LOAD zastępujemy przez:

```
LOAD *"VU-CALC1"SREEN$: LOAD *"VU-CALC2"CODE: GOSUB 9990
```

Pozostaje zapisać program na dyskietce – dokonujemy tego za pomocą instrukcji: **SAVE *"VU-CALC" LINE 10**. Przenosimy drugi blok (SCREEN$):

```
10 LOAD " " SCREEN$
20 SAVE *"VU-CALC1" SCREEN$
```

Przenosimy trzeci blok (typ B) uruchamiając krótki program:

```
10 LOAD " " CODE 25232, 5270
20 SAVE *"VU-CALC2" CODE 25232, 5270
```

Jeśli wykonamy teraz instrukcję **LOAD *"VU-CALC"**, zostanie uruchomiony program VU-CALC z dyskietki. Pozostaje jeszcze ewentualne dołączenie go do zaproponowanego programu MENU.

rys. Piotr Kakiet





Andrzej Izworski, Ryszard Tadeusiewicz



Komputer dla medyka [5]






Uważni Czytelnicy poprzednich odcinków publikowanego cyklu dostrzegli już zapewne charakterystyczną asymetrię przedstawionych dotychczas metod statystycznych: dla danych ilościowych proponowano bowiem korzystanie z testu t-Studenta, natomiast dla jakościowych rekomendowano test Chi-kwadrat. Rozpoczęcie prezentacji zastosowań mikrokomputera w pracy zawodowej lekarza od tych właśnie metod statystycznych było w pełni uzasadnione. Z doświadczenia wiadomo, że w ponad 70% ogółu stawianych w biologii i medycynie zadań obliczeniowych można zastosować jeden z wymienionych testów.

Łatwo jednak zauważyć, że przedstawione metody nie są równoważne. O ile bowiem test t-Studenta pozwala stwierdzić, czy między rozważanymi grupami danych ilościowych występują istotne różnice, o tyle test Chi-kwadrat bada, czy pomiędzy rozważanymi danymi jakościowymi istnieje współzależność. To oczywiście nie to samo, zatem dotychczas umiemy prowadzić inne analizy dla danych ilościowych, a inne dla danych jakościowych – co nie zawsze może być wygodne.

Wprawdzie zawsze można tak sformułować zadanie, aby (dla danych ilościowych) przez wykazanie różnic dało się udowodnić istnienie interesującego związku. Można także tak ustawić badanie związku cech jakościowych, by stwierdzić istnienie w nich interesujących nas różnic. Jednak takie "okrężne" docieranie do prawdy naukowej może się okazać uciążliwe, dlatego teraz uzupełnimy poznane dotychczas metody o technikę wykazywania (wprost) związku pomiędzy danymi ilościowymi. Jeden z dalszych odcinków poświęcimy problemowi badania istotności różnic między danymi jakościowymi.

Głównym zagadnieniem tego odcinka jest wyznaczanie współczynnika korelacji. Pojęcie korelacji jest bardzo często używane (i nadużywane...) w naukach przyrodniczych, przeto celowe będzie poprzedzenie tych rozważań ostrzeżeniem. Otóż obliczany za pomocą dziś wprowadzonego programu współczynnik jest miarą siły związku szczególnego rodzaju – a mianowicie liniowego. Oznacza to, że pomiędzy rozważanymi danymi może występować związek, a mimo to korelacji nie będzie, gdyż będzie to związek nieliniowy. Na przykład: wystarczy, aby jedna z rozważanych danych była proporcjonalna do kwadratu drugiej – a obliczenia wykażą brak znamiennej korelacji, chociaż związek (w sensie przyczynowo – skutkowym) oczywiście jest. Z braku korelacji nie wynika więc niezależność rozważanych danych – chociaż często tak się mylnie wnioskuje.

Stosowanie korelacji wymaga też ostrożności przy formułowaniu wniosków w drugą stronę: otóż jeśli uda się wykazać istnienie nawet bardzo silnej korelacji, to nie wynika z tego, że między rozważanymi danymi zachodzi związek przyczynowy. Możliwe jest bowiem, że istnieje pewien nieznany czynnik, będący wspólną przyczyną obydwu skorelowanych ze sobą zjawisk, opisanych badanymi danymi.

Wyliczany przez program współczynnik korelacji jest liczbą, która może przybierać wartości od -1 do +1. Im większa jest bezwzględna wartość współczynnika korelacji – tym silniejszy związek wykryliśmy. Tak więc wartości współczynnika w pobliżu +1 lub -1 oznaczają istnienie bardzo silnej korelacji, zaś wartości w pobliżu zera można traktować jako dowód jej braku. Natomiast znak współczynnika korelacji oznacza charakter wykrytego związku: dodatni oznacza, że wzrastającym wartościom pierwszej z rozważanych danych towarzyszy wzrost drugiej danej, ujemny zaś oznacza odwrotną prawidłowość – wzrost jednej danej wiąże się z maleniem drugiej.

Współczynnik korelacji jest przydatną liczbą: pozwala on między innymi porównywać siłę związku jednej pary danych z siłą związku innej pary. Można mówić o większym lub mniejszym stopniu skorelowania określonych danych,

> 18

```
1000 LW=6:? A$
1080 ? "6. KORELACJA PROSTOLINIOWA"
1540 ON F GOSUB 10000,15000,
    16000,17000,18000,15500
15500 ? A$:? " KORELACJA PROSTOLINIO-
    WA":? :? :?
15510 GOSUB 23000:IF PF=1 THEN RE-
    TURN
15520 RK=CV:RM=SQR((Q(N1)-
    S(N1)*S(N1)/LN)*(Q(N2)-S(N2)*
    S(N2)/LN))
15522 IF RM=0 THEN ? :? :? "OBLICZENIA
    NIEMOZLIWE":GOSUB
30000:RETURN
15523 RK=RK/RM
15525 NI=LN-2:IF ABS(RK)=1 THEN 15540
15527 T=SQR(NI)*ABS(RK)/SQR(1-RK*RK)
15530 TT=T:GOSUB 21000
15540 ? :? :? "NR-Y ZMIENNYCH ";
    N1,N2
15550 ? " LICZEBNOSCI ";L(N1),L(N2)
15560 ? :? "KOWARIANCJA ";CV
15570 IF CV=0 THEN ? "ZMIENNE
    NIESKORELOWANE":GOTO 15620
15580 ? "ZMIENNE SKORELOWANE"
15590 ? :? "WSP. KORELACJI ";RK
15592 IF RK<0 THEN ? "KORELACJA UJE-
    MNA"
15594 IF RK>0 THEN ? "KORELACJA DO-
    DATNIA"
15600 ? "ISTOTNA KORELACJA ";
15605 IF ABS(RK)=1 THEN 15620
15610 IF PR<0.95 THEN ? "NIE WYSTEPU-
    JE"
15620 GOSUB 30100:RETURN
23000 ? "PODAJ NR 1-SZEJ ZMIENNEJ (1-
    ";K;") ";:GOSUB 30500
23010 IF PF=1 THEN 23000
23020 N1=VAL(P$):IF N1<1 OR N1>K OR
    L(N1)=0 THEN 23000
23025 IF L(N1)<3 THEN ? A$:POSITION
    9,12:? "ZA MALA LICZEBNOSC!":GO-
    SUB 30000:PF=1:RETURN
23030 IF R(N1)<>1 THEN GOSUB
    30300:GOSUB 30100:PF=1:RETURN
23040 ? "PODAJ NR 2-GIEJ ZMIENNEJ (1-
    ";K;") ";:GOSUB 30500
23050 IF PF=1 THEN 23040
23060 N2=VAL(P$):IF N2<1 CR N2>K OR
    L(N2)=0 OR N2=N1 THEN 23040
23070 IF R(N2)<>1 THEN GOSUB
    30300:GOSUB 30100:PF=1:RETURN
23080 IF L(N1)<>L(N2) THEN ? :? :? "ROZ-
    NE LICZEBNOSCI ZMIENNYCH":
    GOSUB 30000:PF=1:RETURN
23090 POKE 621,255:LN=L(N1)
23100 S(N1)=0:S(N2)=0:Q(N1)=0:Q(N2)
    =0:XY=0
23110 FOR I=1 TO LN
23120 A=X(I,N1):B=X(I,N2)
23130 S(N1)=S(N1)+A:S(N2)=S
    (N2)+B:XY=XY+A*B
23140 Q(N1)=Q(N1)+A*A:Q(N2)
    =Q(N2)+B*B
23150 NEXT I
23160 CV=XY-S(N1)*S(N2)/LN
23170 RETURN
30300 ? A$:? "BLEDNY TYP ZMIENNEJ - JA-
    KOSCIOWA!":GOSUB 30000:RETURN
10 GRARHICS 0:W=200:K=5
20 DIM A$(1),E$(1),P$(15):
    A$=CHR$(125)
30 DIM X(W,K),L(K),S(K),Q(K),R(K)
40 ? A$:? " KORELACJA - WPROWADZA-
    NIE DANYCH":?
45 GOSUB 20000:? A$
47 IF R(N) <>1 THEN GOSUB 30300: GOTO
    40
48 IP=1:GOSUB 20500
50 ? A$:POSITION 3,23:? " DANE INNEJ
    ZMIENNEJ (T>TAK, N>NIE)";:INPUT
    F$
60 IF F$="T" OR F$="t" THEN 40
70 GOSUB 15500
75 POSITION 3,23:? "KORELACJA INNEJ
    PARY ZMIENNYCH?"
80 POSITION 12,23:? "(T>TAK, N>NIE)
    ";:INPUT F$
85 IF F$="T" OR F$="t" THEN 40
90 IF F$="N" OR F$="n" THEN ? A$:CLR
    :END
95 ? A$: GOTO 75
20000 ? :? "NUMER ZMIENNEJ (1-";K;")
    ";:GOSUB 30500
20002 IF PF=1 THEN 20000
20005 N=VAL(P$):IF N<1 OR N>KTHEN
    20000
20010 ? :? "TYP ZMIENNEJ (1-ILOSC.,2-JA-
    KOSC.)";:INPUT F$
20012 IF F$<>"1" AND F$<>"2" THEN
    20010
20015 R(N)=VAL(F$):IF F$="1" THEN
    20030
20017 ? :? "LICZBA POZIOMOW ZMIENNEJ:
    ";:GOSUB 30500:IF PF=1 THEN 20017
20018 GOTO 20025
20025 R(N)=VAL(P$):IF R(N)<2 THEN
    20017
20030 ? :? "LICZBA POMIAROW
    (OBIEKTOW): ";:GOSUB 30500
20032 IF PF=1 THEN 20030
20035 P=VAL(P$):IF P<1 OR P>W THEN
    20030
20040 L(N)=P:RETURN
20500 POSITION 21,3:? "POPRAWKI NA
    KONCU!":?
20505 FOR I=IP TO L(N)
20510 ? "POMIAR NR ";I;" :";:GO-
    SUB 30500:IF PF=1 THEN 20510
20512 IF R(N)>1 AND VAL(P$)>R
    (N) THEN 20510
20515 X(I,N)=VAL(P$)
20520 NEXT I
20530 ? A$:POSITION 2,10:? "KONIECZNE
    POPRAWKI? (T>tak,N>nie)";:IN-
    PUT F$
20540 IF F$="N" OR F$="n" THEN RE-
    TURN
    20545 IF F$<>"T" AND F$<>"t"
    THEN 20530
20550 ? A$:? :? "POPRAWIANY POMIAR
    NUMER ";
20560 GOSUB 30500:IF PF=1 THEN 20560
20565 I=VAL(P$):IF I<IP OR
    I>L(N) THEN 20550
20570 ? :? " WARTOSC ";:? X(I,N)
20580 ? "NOWA WARTOSC ";:GOSUB
    30500:IF PF=1 THEN 20580 20585 IF
    R(N)>1 AND VAL(P$)> R(N) THEN
    20580
20590 X(I,N)=VAL(P$):GOTO 20530
21000 PR=1:Y=1:T=T^2:V=Y:E=NI:Z=T
21010 J=2/9/V:C=2/9/E:LP=ABS((1-C)
    *Z^(1/3)-1+J)/SQR(C*Z^(2/3)+J)
21020 IF E<4 THEN LP=LP*(1+0.08*LP^4/
    E^3)
21030 PR=0.25/(1+LP*(0.196854+
    LP*(0.115194+LP*(3.4E-04+LP*
    0.019527))))^4
21040 PR=2*PR:IF T>1 THENPR=1-PR
21050 RETURN
30000 POKE 621,255:FOR I=1 TO500:NEXT
    I:POKE 621,0:RETURN
30100 POKE 621,0:POSITION 12,23:? "NACIS-
    NIJ RETURN";:INPUT F$:? A$:RETURN
30500 INPUT P$:PF=0:PK=0:IF P$ =" "
    THEN 30550:PP=ASC(P$(1,1))
30510 FOR J=(PP=43 OR PP=45)+1 TO
    LEN(P$):PW=ASC(P$(J,J))
30520 IF PW=46 THEN PK=PK+1:IF PK>1
    THEN 30550
30530 IF PW<46 OR PW=47 ORPW>57
    THEN 30550
30540 NEXT J:RETURN
30550 PF=1:SOUND 0,110,10,10:FOR J=1 TO
    50:NEXT J:SOUND 0,0,0,0:RETURN
```

17 <

prowadzić głębsze rozważania na temat przypuszczalnej ważności różnych przyczyn składających się na określone zjawisko itd. Zwykle jednak dla użytkownika programu zasadnicze znaczenie ma odpowiedź na podstawowe pytanie: Jest związek czy go nie ma?

W praktyce nigdy nie uzyskuje się współczynnika korelacji równego 1 lub -1, nawet dla ewidentnie powiązanych ze sobą obserwacji, podobnie jak nigdy nie wylicza się współczynnika dokładnie wynoszącego 0, nawet dla danych absolutnie nie mających ze sobą związku. Wynika to z faktu, że dane pochodzą z obserwacji i na ich wartości nakładają się rozmaite czynniki przypadkowe – warunkujące zresztą celowość odwoływania się do statystyki. Orzekanie, czy korelacja jest istotna, czy nie – musi być oparte na specjalnym teście. Szczegóły tego testu są dość złożone, nie musimy tu w nie wnikać – wystarczą odpowiednie reguły zawarte w programie.

Jak zwykle w tym cyklu, programy wykonujące opisane obliczenia przedstawiono w dwóch wersjach. Pierwsza, krótsza - przeznaczona jest dla stałych Czytelników, którzy dopisując pokazane w pierwszym tabulogramie czterdzieści wierszy do utworzonego w poprzednich odcinkach programu, mogą go wzbogacić o możliwość wyliczania i oceny współczynnika korelacji. Po podaniu numerów zmiennych, dla których mają być wykonane obliczenia, kontrolowana jest poprawność postawionego zadania przez sprawdzenie, czy wskazane zmienne są typu ilościowego (gdyż tylko dla tego typu danych wolno liczyć korelację!). Ponadto sprawdzany jest warunek jednakowej liczebności danych dla obydwu korelowanych zmiennych. Warunek ten jest konieczny, ponieważ podlegające korelacji dane – umówmy się na chwilę nazywać je X i Y – powinny tworzyć pary obserwacji: $(x_1, y_1), (x_2, y_2),...,(x_L, y_L)$. Jeśli korelacja ma być wykazana, to w tych parach powinno się dostrzegac wpływ wartości x_i na odpowiadającą jej wartość y_i. Bez takiego przyporządkowania danych nie ma sensu badać korelacji – a dla danych stanowiących pary oczywiście liczebność obserwacji X i Y musi być taka sama!

Druga, dłuższa wersja programu przeznaczona jest dla osób, które interesuje program obliczania korelacji jako oddzielny moduł. Muszą one wpisać cały tekst z drugiego tabulogramu (zaczynający się od wiersza z numerem 10) oraz dopisać do niego tekst z pierwszego tabulogramu pomijając wiersze o numerach 1000, 1080 i 1540.

Zasady posługiwania się autonomicznym programem liczącym korelacje są identyczne, jak wyżej omówione zasady korzystania ze wstawki do globalnego programu.

W następnym odcinku obliczenia korelacji wzbogacone zostaną o równanie regresji.

■

\W domu\

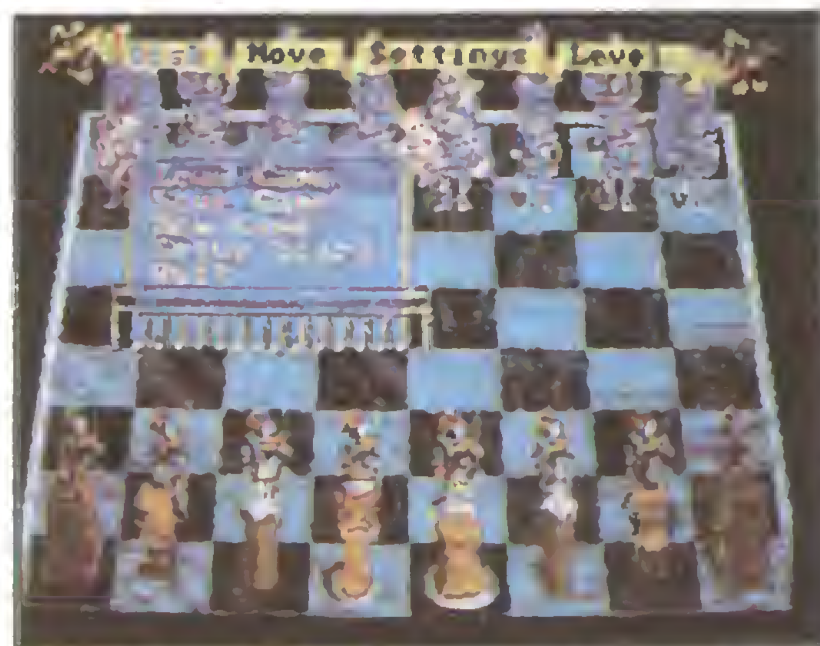
Przemysław Wnuk



Gry do peceta [2]

Nazwa: *Battle Chess*
Autor: *Interplay*
Komputer: *IBM PC 512K*

Battle Chess jest programem szachowym oferującym oprócz standardowych możliwości spotykanych w większości podobnych programów także wzbogaconą grafikę. Gra może toczyć się na zwykłej planszy przy użyciu klasycznych pionków, lub też na trójwymiarowej szachownicy z figurami, przedstawiającymi postacie w średniowiecznych strojach. Figury są wtedy animowane, a każdorazowe zbicie poprzedzone jest krótkim, efektownym pojedynkiem. Program umożliwia grę przeciw komputerowi przy dziesięciu poziomach jego "inteligencji" oraz ograniczenie czasu przeznaczonego na wykonanie ruchu. Dostępne są także opcje gry dla dwóch partnerów, cofania ruchu, zamiany stron, podpowiedzi itp., słowem – wszystko to, co potrzeba do gry lub nauki gry w szachy. Na szczególną uwagę zasługuje wspomniana oprawa graficzna. Autorzy dużo wysiłku włożyli w staranną animację figur. Poruszają się one zgodnie ze swoją rolą (np. królowa dyskretnie kręci biodrami, wieża zamienia się w ociężałego małpoluda). Najciekawszy i najdowcipniejszy jest jednak pojedynek figur poprzedzający zbicie. Każdy pion walczy wówczas dostępną dla siebie bronią – biskup (czyli nasz goniec) atakuje pastorałem, wieża boksuje, a królowa rzuca czary zamieniające przeciwnika w szkielecik.



W trybie trójwymiarowym gra jest, mimo licznych atrakcji, dość niewygodna, ze względu na pewne trudności przy rozróżnieniu figur spowodowane bogactwem strojów. Wrażenie to pogłębia fakt, że ich wygląd dostosowany jest do angielskiego nazewnictwa – goniec jest biskupem, koń – rycerzem, Szachiści, pragnący skupić się jedynie na grze, będą preferować zapewne dwuwymiarową szachownicę.

Battle Chess godny jest polecenia zarówno zaawansowanym graczom, jak również początkującym adeptom sztuki szachowej z uwagi na duże możliwości kontroli i zmiany przebiegu gry. Program nadaje się do wykorzystania także przy demonstracji bogactwa graficznego IBM PC. Gra dostarczana jest na dwóch dyskietkach nie zabezpieczonych przed kopiowaniem (program zadaje na wstępie pytanie o przebieg partii szachowej opisanej w podręczniku). Dołączony jest do niej krótki podręcznik dla początkujących, prezentujący pokrótce zasady gry w szachy i najprostsze strategie. Program wymaga minimum 512K RAM.

Nazwa: *Skate or Die*
Autor: *Electronics Arts*
Komputer: *IBM PC*

Skate or Die jest symulatorem popularnej u nas swego czasu deskorolki. Grę zaczynamy w sklepie, gdzie wybieramy odpowiedni kolor sprzętu i decydujemy, czy zaczniemy skromnie od próby naszej sprawności, czy też od razu przystąpimy do szlachetnej rywalizacji. Następnie wyjeżdżamy ze sklepu i kierujemy się w którąś z uliczek



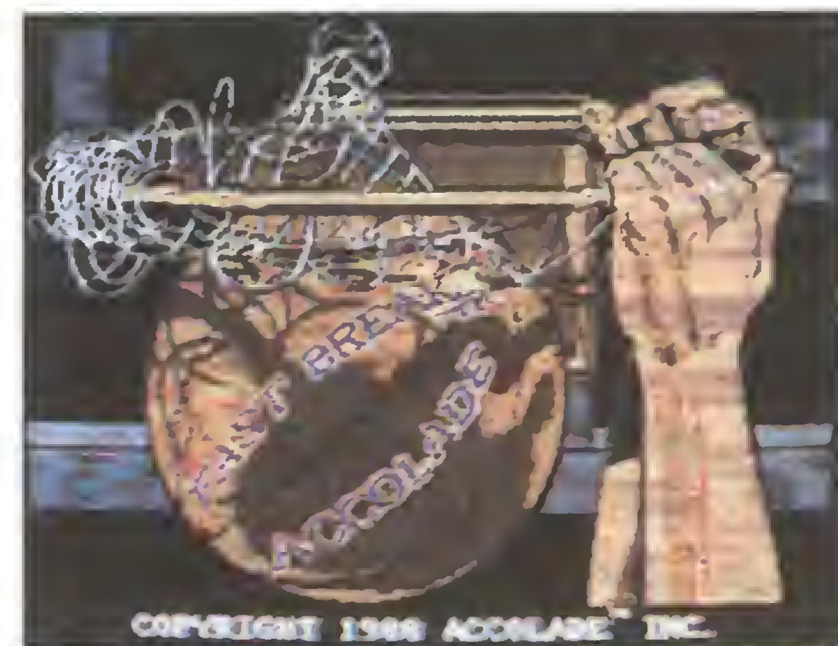
prowadzących do miejsc rozgrywania poszczególnych konkurencji. Do wyboru mamy: wyścigi na torze lub uliczkach slumsu, rampę do skoku wzwyż lub akrobacji oraz specjalny basen, gdzie toczymy sprawdzamy umiejętności jeździeckie wspomagane specjalną tyczką walcząc z przeciwnikiem. Możemy wykonywać różne ewolucje, skoki, obroty i skręty, za które przyznawane są punkty. W czasie wyścigów bardzo skuteczne jest taranowanie przeciwnika, jednakże wymaga to zręczności, aby samemu się nie przewrócić. Najwięcej emocji dostarczają jednak ewolucje na rampie i walka w basenie, której celem jest trzykrotne trafienie przeciwnika tyczką. Możemy wybrać trzech oponentów, z których najgroźniejszy i zarazem "najurodziwszy" jest niejaki Lester. Manewruje on deskorolką po mistrzowsku i trafienie go tyczką jest rzeczywiście nie lada sztuką.

Program wymaga komputera XT z 256K RAM oraz karty CGA lub EGA, jednakże optymalną szybkość uzyskać można na AT. Zalecany byłby także twardy dysk, w razie jego braku istnieje konieczność częstej zmiany dyskietek. Program nie jest zabezpieczony przed kopiowaniem, konieczne jest natomiast posiadanie specjalnej wkładki z kodami, załączonej do oryginału.

W sumie jest to typowa gra zręcznościowa przeznaczona dla tych miłośników akrobacji sportowej, których natura wyposażyła w zręczne palce, a rodzice – w komputerową mysz.

Nazwa: *Fast-Break*
Producent: *Accolade*
Komputer: *IBM-PC*

Jeśli jeszcze nie nauczyłeś się grać w koszykówkę, masz szansę nadrobić ten brak uruchamiając program *"Fast-Break"*. Program posiada kilka opcji, w tym doskonalenie różnych taktycznych wariantów gry (atak na kosz, obrona w tzw. trumnie) opisanych wraz z diagramami w załączonym podręczniku. Sama gra jest, moim zdaniem, nieco mniej ciekawa, niż mecz koszykówki oglądany na



żywo, jednakże osoby lubiące gry zręcznościowe mogą być innego zdania. Jej sens polega na szybkim przekazywaniu piłki od gracza do gracza w celu wypracowania pozycji do ataku na kosz, zatem zręczność odgrywa przy tym główną rolę.

Program dostarczany jest na jednej dyskietce nie zabezpieczonej przed kopiowaniem. Sterowanie odbywa się za pomocą klawiatury, jednakże mysz znacznie polepsza uzyskiwane wyniki. Dyskietka zawiera również *drivery* dla trzech podstawowych standardów graficznych.

Opisywane gry otrzymaliśmy od firmy Electronics Arts Ltd., 11-49 Station Road, Langley, Berks. SL3 8YN, Anglia.

■





Krzysztof Matey



Nasze programy

Program: ***ODDZIAŁ COBRA (Strike Force Cobra)***
Producent: ***Redakcja Programów Komputerowych WWP, licencja: Macmillan Ltd. England***
Rok produkcji: ***1988***
Komputer: ***ZX Spectrum***
Cena: ***650 zł***

Stało się! Ludzkość przerażona jest nuklearnym szantażem szalonego geniusza, znanego jako WRÓG. Ma on system komputerowy umożliwiający przejęcie kontroli nad całym blokiem obronnym Świata. WRÓG stawia żądania i grozi zniszczeniem Świata. Dla swoich celów WRÓG porwał czołowych informatyków i groźbami tortur zmusił ich do pracy. Są oni uwięzieni w twierdzy, labiryncie, głównej bazie WROGA. Każdy porwany zna jedną cyfrę szyfru potrzebnego do otwarcia pancernych drzwi do pomieszczenia głównego komputera. Jeżeli ktoś chciałby zniszczyć komputer WROGA, koszmar zacznie się automatycznie.

Mając alternatywę zniszczenia lub niewoli, Rada Światowa zdecydowała się przeciwdziałać. Z najlepszych fachowców stworzono oddział o kryptonimie "Cobra", który musi dostać się do twierdzy WROGA, unieszkodliwić systemy obronne oraz zlokalizować i wyłączyć komputer główny.

Naczelne Dowództwo zadecydowało, że największe szanse ma czteroosobowa grupa. Tobie – grającemu – powierzono wybór odpowiednich osób i dowodzenie całą wyprawą. Wiesz, że wszyscy komandosi są równie sprawni i mają te same umiejętności. Tylko czterech może wziąć udział w akcji. Musisz mieć jednak świadomość, że wybór nie ma wpływu na przebieg misji.

Grupa została skompletowana. Wybrani komandosi oczekują na odprawę oddziału w Tajnym Ośrodku Specjalnym Naczelnego Dowództwa. W niewielkim pokoju, sali odpraw, zebrali się już wszyscy. Na ścianie plan jakichś pomieszczeń. Jeszcze chwila i odprawa się zaczyna. Głos zabrał komendant tajnego ośrodka. Oto co usłyszeli uczestnicy wyprawy:

Witam. Zadanie jest wam znane. Za wszelką cenę wyłączyć komputer główny WROGA. Nie dopuścić do nuklearnej katastrofy. Do realizacji tego zadania zostaliście wybrani właśnie wy. Oto znane nam informacje, dostarczone przez wywiad Naczelnego Dowództwa.

Cel misji, komputer główny, znajduje się na czwartym poziomie. Nim do niego dotrzecie, musicie pokonać wszelkie niespodzianki. Dostępu bronią różne pułapki oraz roboty bojowe i uzbrojeni strażnicy. Czas misji jest ograniczony. Wasze wejście spowoduje rozpoczęcie wstecznego odliczania czasu do momentu uruchomienia broni WROGA. W niektórych pomieszczeniach znajdują się pomocnicze komputery organiczne. Należy je zniszczyć. Eliminacja ich spowoduje wydłużenie czasu odliczania, a co za tym idzie i czasu waszej misji. Zostaliście wyposażeni w wytrych cyfrowy DLB, jest on niezbędny do przejścia przez drzwi prowadzące do pomieszczenia głównego komputera. Potrzebne cyfry kodu znają uwięzieni naukowcy. Musicie ich znaleźć. Aby otworzyć drzwi, trzeba znać co najmniej sześć cyfr kodu. Przy mniejszej liczbie cyfr DLB może rozszyfrowywać za długo. Wasze uzbrojenie to pistolety laserowe i granaty. Jeżeli ktoś z oddziału zostanie ranny (ma mało energii), powinien skorzystać z apteczki. Znajdują się one w niektórych pomieszczeniach. Jeśli znajdziecie się przed drzwiami należy je otworzyć: kopniakiem, stając na specjalnej płytce naciskowej lub dźwignią z konsoli sterującej. Konsole sterujące drzwiami są oznaczone literą D. Między poziomami można się poruszać jedynie za pomocą wind. Windy sterujemy z konsol oznaczonych L. Przełączenie dźwigni następuje przez wskoczenie na pudło znajdujące się przy konsoli.

Musicie być bardzo ostrożni. Wszelkie ruchome przedmioty są dla was bardzo niebezpieczne. Możecie napotkać strażników, roboty bojowe, lasery, bloki pod wysokim napięciem, automatyczne działka i karabiny oraz latające talerze. Latające talerze można niszczyć granatami lub celnymi strzałami z pistoletu laserowego, najlepiej z wyskoku. Pozostałych urządzeń nie można zniszczyć. Trzeba jedynie zakłócić ich pracę strzelając do nich lub rzucając granatami. Powoduje to np. spowolnienie ruchów robotów czy strażników. Każde takie zakłócenie należy wykorzystać. Pragnę zaznaczyć, że okna nie są zamknięte. Można przez nie wskakiwać. Tam gdzie jeden z was nie będzie sobie radził, niech pomoże mu inny. Wyższe przeszkody pokonujecie, wchodząc jeden na drugiego.

Tyle uwag ogólnych. Przejdźmy do planu opanowania pierwszego poziomu. Proszę spojrzeć na szkic. Oznaczenia literowe są następujące: DC – konsola otwierająca drzwi, LC - konsola uruchamiająca windę, L – winda, H – naukowiec.

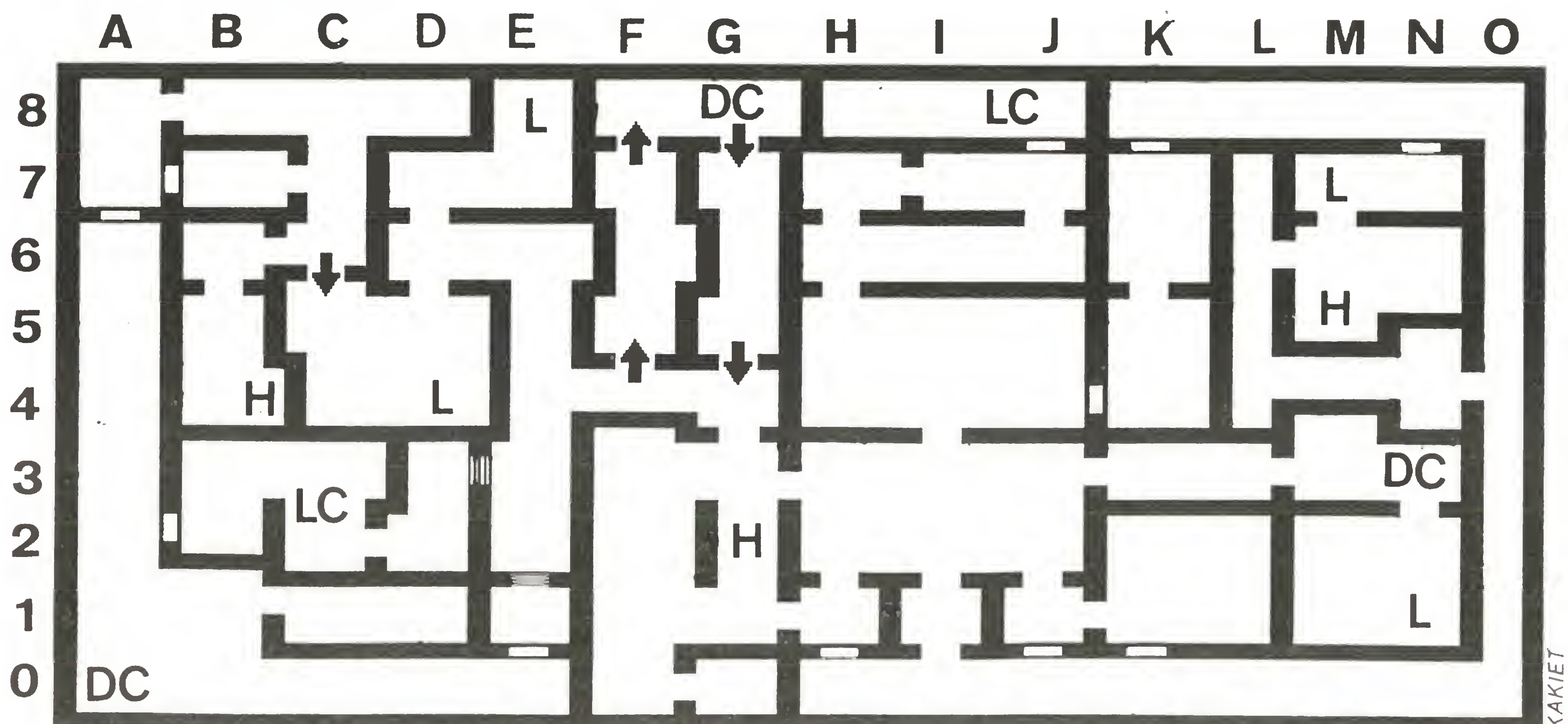
Zapoznam was z pozycjami wyjściowymi. Komandos 1. znajduje się na O4. Znajdujące się tu drzwi są zamknięte. Komandos 2. znajduje się na O8, a komandos 3. na A6. Komandos 4. stoi przy oknie, na E0. Wewnątrz czuwa patrolujący robot.

Waszym zadaniem na tym poziomie jest wejście do wnętrza, opanowanie kondygnacji i zajęcie dogodnej pozycji do ataku na poziom drugi.

Teraz zadania szczegółowe dla każdego z was.

Słuchajcie komandosie 3. Z A6 przejdziecie na A2. Wskoczycie przez okno na B2. Uważajcie na bloki pod wysokim napięciem. Omijając strażnika na skuterze, zajmiecie dogodną pozycję do zniszczenia latającego talerza i zestrzelicie go. W sterowni windy na C2 wskoczycie na blok przy dźwigni. Uruchomi to windę na N1. Pójdziecie dalej na D2, otwierając kopniakiem drzwi. Przeskoczycie z D3 na

> 20



19 <

E3 przez ruchomą żaluzję. Uważajcie na robota z E3. Przejdziecie na E2 i staniecie na płytce naciskowej przy podnoszonych drzwiach. Drzwi otworzą się. Przesuniecie się na bok, wciąż stojąc na płytce. Czekacie tutaj. Nie wchodzicie do następnego pomieszczenia.

Teraz komandos 4. Wrzucicie przez okno granaty, co obezwładni na pewien czas robota. Wskoczycie do E1. Poruszajcie się skokami, by uniknąć spotkania z robotem. Wejdziecie drzwami do E2. Tu czeka na was komandos 3. Posuwacie się dalej, unikając robota, staniecie na E3, blisko wież bloków "Z".

Uwaga! Komandosie 3. Idziecie za komandosem 4. w kierunku wież. Stajecie na jego ramionach i przeskakujecie na drugą stronę. Skręcacie na prawo. Otworzycie kopniakiem pierwsze drzwi na lewo. Wejdziecie na F5. Unikajcie kul automatycznego działka. Przeskoczycie promienie lasera. Wskoczycie przez okno do F8 i natychmiast kucniecie. Zestrzelicie latający talerz i uruchomicie urządzenie sterujące otwarciem drzwi na B8, wskakując na blok przy dźwigni. Przeskoczycie do G7, zaraz po tym skręćcie, by nie wylądować w promieniach lasera. Posuwacie się w kierunku G4, unikając robota na G5. Wejdziecie w drzwi na G3 i starannie wybierając moment, przeskoczycie nad ruchomą przeszkodą. Apteczkę znajdziecie w pomieszczeniu G0. Drzwi otwieracie kopniakiem. Wracacie do zakładnika – naukowca na G2. Ujawni on wam część kodu cyfrowego dla DLB. Gdy wskoczycie mu na ramiona, pomoże wam przeskoczyć wieżę "Z". Z G3 przejdziecie korytarzem (G4-E4-E6) zestrzeliwując lub omijając latający talerz. Z D6 przez drzwi wejdziecie na D5. Zestrzelicie następny latający talerz i staniecie przy windzie na D4. Czekacie tutaj.

Znowu wy, komandosie 4. Unikając robotów wrócicie na E0. Przejdziecie na B0. Tutaj trzeba przykucnąć. Zestrzelicie latający talerz. Na A0 przesuniecie dźwignię urządzenia otwierającego drzwi. Następnie zniszczycie pomocniczy komputer organiczny na D1. Skierujecie się na A6. Przeskoczycie przez okno na A7. Przejdziecie przez drzwi na B8. Zestrzelicie latający talerz. Apteczka jest ukryta na D8 (przedni prawy róg). Aby się do niej dostać, musicie skoczyć. Dla ułatwienia tego komandos 3. może się przesunąć na D7. Uważajcie na roboty. Wejdziecie do B6, a następnie do B5. Przeskoczycie ostrożnie obok robota. Naukowiec z B4 poda wam drugą cyfrę szyfru. Udacie się do C5 i korzystając z bloku w rogu, przeskoczycie przez przeszkody. Ustawicie się przy windzie na D4.

Kolej teraz na komandosa 1. Przesuniecie się wzdłuż ścian budynku w kierunku okna przy J0. Wrzucicie granat do J1. Wejdziecie i staniecie na płytce naciskowej. Otworzą się podnoszone drzwi do K1 i K3. Czekacie w tym miejscu.

Zajmijmy się teraz komandosem 2. Przesuniecie się w lewo na K8. Wskoczycie przez okno. Cofniecie się, by nie wpaść na stos bloków. Kucniecie (uwaga na karabin) i szybko przeskoczycie przez dwa rzędy bloków. Otworzycie kopniakiem drzwi przy K6 i wejdziecie ostrożnie do K5. Poruszajcie się bardzo ostrożnie. Przeskoczcie do J4. Omijając karabiny i roboty przejdziecie do J6. Wejdziecie do J7, a potem do J8. Przeskoczycie ponad ruchomą przeszkodą i włączycie mechanizm windy – D4 dla komandosów 3. i 4. Wrócicie do I5 tą samą drogą, którą przyszliście. Przeskoczycie szybko przez lasery do I3. Przejdziecie do I0 i staniecie pod oknem na H0. Apteczka jest na H1. Przez okno zestrzelicie latający talerz. Wtedy skorzystajcie z apteczki. Uważajcie, nie możecie wejść na płytkę naciskową. Wrócicie na J3 i przejdziecie K3. Drzwi są otwarte. Idźcie za skuterem po linii kwadratu zgodnie z ruchem wskazówek zegara. Strzelając wejdziecie do M3. Spróbujcie zatrzymać skuter ze strażnikiem, by dostać się do N3, do dźwigni sterującej drzwi. Przesuniecie dźwignię. Urządzenie otwiera drzwi na O4 i M7. Wejdziecie ostrożnie do N2. Pójdziecie w lewo. Kiedy śmignie kula, skierujecie się do rogu. Skokami omijajcie kolejne kule. Jeżeli po tym wszystkim będziecie jeszcze cali, to staniecie przy windzie.

Wasza kolej, komandosie 1. Wyskoczycie oknem do J0 i wejdziecie oknem przy K0 do K1. Na L2 znajduje się pomocniczy komputer organiczny. Zniszczycie go. Wrócicie na K0. Przejdziecie wzdłuż budynku do drzwi przy O4 i wejdziecie do N4. Pójdziecie korytarzem do L6 niszcząc latający talerz. Apteczkę znajdziecie w L7. Rozwalicie drzwi przy M6, wejdziecie rzucając dwa, trzy granaty. Robot zwolni swój ruch i łatwo dotrzecie do uwięzionego naukowca. Otrzymacie informacje o następnej cyfrze szyfru. Rzucicie jeszcze jeden granat i wejdziecie do windy przy M7.

Jeżeli każdy z was dokładnie wykona swoje zadania, to poziom zostanie opanowany. Będziecie też w posiadaniu trzech cyfr kodu DLB. Wszyscy czekacie przy windach, komandosi 3. i 4. na D4, komandos 1. na M7, komandos 2. na M1.

Po zjechaniu windami na poziom drugi czeka was opanowanie pozostałych kondygnacji bazy. Niestety wywiad Naczelnego Dowództwa nie zdobył informacji o tych poziomach.

Powodzenie akcji zależy od waszej szybkości, odwagi i dobrego planu. Musicie liczyć tylko na siebie. Mam nadzieję, że doświadczenie, jakie zdobędziecie na pierwszym poziomie, pozwoli łatwiej pokonywać następne trudności.

Czy są pytania? Nie ma, w takim razie życzę powodzenia.

Jeśli chcesz sprawdzić czy dane wywiadu były prawdziwe, to spróbuj zagrać w tę grę.



Atari XE/XL

Tomasz Belowski z Olsztyna do poprawiania gier wykorzystuje program dyskowy SHERLOCK. A oto efekty jego pracy:

CISIN'COUSIN

LDA #05 hex A905

STA 6825 hex 8D2568

Należy zmienić wartość LDA na FF wówczas otrzymamy 255 ludzików. (Należy zabić pierwszą piątkę).

PANTHER

LDA #05 hex A905

STA A0 hex 85A0

Limit błędów znajduje się w instrukcji LDA.

ARCANOID

LDA #04 hex A904

STA 4D hex 854D

Limit błędów nie może wynosić 255, ponieważ zdobycie dodatkowego ludzika powoduje "przekręcenie" licznika.

DAN STRICKE BACK

DEC 0608 hex 0806

Instrukcja DEC powoduje zmniejszenie licznika o jeden. Należy ją zmienić na EE (zwiększenie wartości o jeden) lub na EA (nic nie rób).

BOULDER DASH 4

DEC 07B6 hex CEB607

Jak wyżej.

Krzysztof Kołodziej z Katowic dzieli się swoimi spostrzeżeniami na temat niektórych gier:

PRELIMINARY MONTY 16k

Po utracie ostatniego ludzika możemy kontynuować grę naciskając przycisk FIRE. Zyskujemy w ten sposób 5 nowych istnień.

LASER HAWK

Można w niej otrzymać "nieśmiertelność" po wciśnięciu klawisza B przed uruchomieniem programu. Po przejściu wszystkich plansz, by wrócić do normalnego trybu gry, trzeba ponownie nacisnąć klawisz B.

MIRAX FORCE

"Nieśmiertelność" uzyskujemy przez wciśnięcie klawisza C (powoduje chwilowe "zawieszenie się" komputera). Następnie naciskamy wszystkie klawisze od ESC do M włącznie. Klawisz M uruchomi grę. Możemy ją przerwać naciskając ESC. Do normalnego trybu gry powracamy powtarzając tę samą czynność co poprzednio.

Robert Wróblewski, uczeń IV klasy L.O. w Inowrocławiu przesyła długą listę poprawionych gier.

THE GOONIES

Gra oparta na opowiadaniu i filmie Spielberga. Jest ona bardzo trudna, a dodatkowym kłopotem jest dogrywanie scen (z magnetofonu). Po przejściu do np. 3. sceny i wyczerpaniu tam limitu błędów (5), trzeba wczytywać od początku sceny, tzn. wczytać 1, przejść, wczytać 2, przejść itd. Zadaniem gry jest ochrona domu naszych bohaterów przed sprzedażą, co w rzeczywistości sprowadza się do przejścia przez labirynt scen. Poprawki zatrzymują odliczanie scen.

Główny blok programu (na który wgrywamy później sceny) składa się z 3 segmentów. Pierwszy kopiujemy bez zmian, natomiast drugi (ok. 20 kB) wgrywamy na *GameCopy* i pod adresami (dec) znajdujemy dwa razy zapisaną liczbę określającą limit błędów.

25270 LDAIM 5 co w tym programie odpowiada LDA #5

25272 STA4396

25287 LDAIM 5 odpowiada LDA #5

25289 STA 4396

oraz

15798 DEC 4396 i 26038 DEC 4396

Rozkazy te "odliczają" wpadki.

Wybieramy (w *GameCopy*) opcję 1 – rozpoczynamy asemblację od adresów:

15798 NOP 26038 NOP

15799 NOP 26039 NOP

15800 NOP 26040 NOP

aby wyjść z opcji 1; po pojawieniu się następnego adresu wciskamy RETURN. Opcję 8 zapisujemy na kasecie. Następnie kopiujemy 3 segment bez zmian. I już można kontynuować przygody bohaterów bez końca.

PRELIMINARY MONTY

Oprócz ulubionego "wiecznego życia" jeszcze jeden sposobik: na pierwszym poziomie, po straceniu 6 ludzi, gdy naciśniemy FIRE, to stracimy wszystkie punkty, ale dostajemy 6 ludzi. Przedmioty, które mieliśmy, nadal zostają w naszej kieszeni.

Oto poprawki:

kopiujemy ładowacz "!", wczytujemy program właściwy (w *GameCopy*) i deasemblujemy:

26026 LDAIM 6 LDA #6

26028 STAZ 136 STA 136 – STAZ używane dla kom. 1-bajtowej,

oraz

27898 DEC 136 DEC 136 – DEC kom. 1-bajtowe,

po wybraniu asemblacji wpisujemy:

27898 NOP

27899 NOP i mamy "wieczne życie".

GREEN BERET

Poprawki dotyczą wersji 45 kB, z czego ok. 15 kB to instrukcja BRK. Po wprowadzeniu poprawek program trzeba skopiować za pomocą Pirata. Zabezpieczy on program, pozbędzie się 15 kB instrukcji BRK, nie robiąc przy tym żadnych szkód. Wgrywamy właściwy program do TURBO COPY i po wybraniu opcji W znajdujemy:

$489Ahex-18486dec – znajdujemy $CE,$CC,$06 DEC$06CC – odliczanie "wpadek",

$45E0hex-17888dec – znajdujemy $A9,$03 LDA #03 – liczba wznowień gry; 0 też jest traktowane jako życie,

$45E2hex-17890dec – znajdujemy $8D,$CC,$06 STA $06CC – wpisujemy kursor na dany adres w pamięci, pod adresem $489Ahex-18486dec wpisujemy $AD tak, że zaczynając od tego adresu mamy ciąg bajtów: $489Ahex-18486dec – $AD,$CC, $06 LDA $06CC

Następnie wychodzimy z tej opcji (W) przez naciśnięcie spacji i pro-

gram nagrywamy na kasetę roboczą. Wgrywamy Pirata, program uruchamia się i "chodzi" bez zarzutu.

MOLECULE MAN

Poprawki związane z ubywaniem pigułek antyradiacyjnych. Pierwszy blok programu (po którym pojawiają się napisy) kopiujemy bez zmian, drugi blok wczytujemy do TURBO COPY i w opcji W znajdujemy:

$08CChex-2252dec-$A9$14 LDA #14 (początkowy zapas pigułek dec 20),
$08CEhex-2254dec-$8D$06$06 STA $0606
oraz
$ODDAhex-3546dec-$CE$06$06 DEC $0606 – odejmowanie pigułek.

Poprawki należy wprowadzić tak, aby pod tymi adresami było:
$ODDA-3546dec-$AD,$06,$06 LDA $0606
$8CC-2252dec-$A9,$01 LDA #1

ZX Spectrum

Arkadiusz Michalik z Pionek, student I roku elektroniki P.W. nadesłał poprawki do 3 gier.

STYX

od adresu #749 następuje wpisanie limitu błędów (początkowo 3)
#749F LD A,03
#74A1 LD (#747E),A
Limit ten możemy zmienić wpisując POKE 29856, nowy limit (0 do 255). Natomiast od adresu #7C10 zmniejszany jest licznik:
#7C10 LD A,(#747E)
#7C13 DEC A
#7C14 JP Z,#7481
#7C17 LD *#747E),A
#7C1A JP #74FO
Jednak wpisanie w komórkę #7C13 zera nic tutaj nie da. Trzeba wpisać zera w komórki od 31763 do 31766 i uzyskamy "wieczne życie".

Wersja składa się z programu ładującego w Basicu, czołówki i dwóch segmentów:

nazwa	długość	start
S2	768	26624
S3	4096	28672

Poprawki wprowadzamy do segmentu S3.

Można też napisać prosty program ładujący:

```
10 CLEAR 24570:LOAD " "CODE: LOAD" "CODE:LOAD" "CODE: POKE 29856, N : RANDOMIZE USR 29696
```

gdzie N jest liczbą możliwych wpadek lub zamiast jednej instrukcji POKE wpisujemy
POKE 31763,0:POKE 31764,0:POKE 31765,0:POKE 31766,0
i wtedy uzyskujemy "nieśmiertelność".

THE BIRDS AND THE BEES (firmy BUG-BYTE SOFTWARE)

Zamieniamy się w pszczółkę. Naszym zadaniem jest zbieranie nektaru z kwiatów i dostarczanie go do ula. W pracy tej przeszkadzają nam ptaki i dżdżownice. Wersja składa się z programu ładującego i głównego segmentu o długości 24650 i adresie 32700. Znajdujemy:
#916A LD A,(#84E9)
#916D SUB #01
#916F JP C,#9178
#9172 LD (#84E9),A
....
#93DB LD A,#04
#93DD LD (#84E9),A

Wpisując POKE 37852, N (N liczba wznowień gry – od 0 do 255) lub POKE 37229,0 uzyskamy "nieśmiertelność".

Można też napisać program ładujący:

```
10 CLEAR 32699:LOAD" "CODE:
poprawki:
  RANDOMIZE USR 32700
```

GLUG – GLUG

Naszym zadaniem jest zbieranie skarbów z dna morskiego i dostarczanie ich na statek. Liczba pięciu wskrzeszeń jest niestety niewystarczająca. Znajdujemy zmniejszanie licznika osobno dla pierwszego i drugiego gracza.
Pierwszy gracz:
34136 LD A,(#729E)
34139 DEC A
34140 LD (#729E),A
34143 CP#00
34145 JR NZ#85A0
Drugi gracz:
34173 LD A,(#729F)
34176 DEC A
34177 LD (#729F),A
34180 CP#00
34182 JR NZ #85A0

"Nieśmiertelność" uzyskamy wpisując:
POKE 34139,0 – dla pierwszego gracza,
POKE 34176,0 – dla drugiego gracza.

Wersja, którą dysponuje Arkadiusz, zaczyna się od adresu 23296 (w języku maszynowym). Przed wczytaniem głównego bloku trzeba więc zmienić rozkaz w programie ładującym z JUMP 34349 na RET, aby można było wprowadzić poprawki. Można również wpisać:

```
10 LOAD"CODE: POKE 23313,201: RAND USR 23297: POKE 34139,0: POKE 34176,0: RANDOMIZE USR 34349
```

i program ten załaduje ekran, główny blok, wprowadzi poprawki i uruchomi grę. Główny segment: adres 24000, długość 11570, start 34349.

Amstrad CPC 464, 664, 6128

Janusz Szymański uczeń I klasy XIV LO im. K. Gotwalda w Warszawie przysłał rozwiązanie "nieśmiertelności" w grze **PYJAMARAMA**. Poprawki dotyczą tej wersji programu, w której blok główny (PYJAMA1.BIN) ładowany jest od adresu 8192 (2000H), o długości wynoszącej 34508 (86CCH). Trzeba zmienić adres 3ED7 wpisując 3E (w miejsce 35) oraz adres 3ED8, wpisując 02 (w miejsce 7E).

Marcin Borowiecki uczeń II klasy L.O. proponuje porawki do gier **JET SET WILLY** i **BLAGGER**.

W grze BLAGGER POKE &7F06,&FF wpisane w program ładujący przed instrukcją CALL &7F56 zwiększa limit pomyłek do 255.

W przypadku JET SET WILLY sprawa się komplikuje. Sama "nieśmiertelność" nie jest wystarczająca. W tej grze bowiem istnieje możliwość dostania się w pułapkę, w której komputer zmniejsza limit błędów do zera w błyskawicznym tempie, po czym rozpoczynamy grę od początku. Innej drogi wyjścia nie ma. Po wpisaniu "nieśmiertelności" komputer nigdy nie zmniejszy limitu do zera i drastycznym, aczkolwiek jedynym skutecznym sposobem wyjścia z pułapki jest wyzerowanie komputera i ponowne załadowanie gry. Aby temu zapobiec, poniższy program zaopatrzony jest w odpowiednią procedurę. Powoduje ona, iż po naciśnięciu klawisza sterowania kursorem ku górze w momencie "zejścia", komputer wyczerpuje limit błędów i przechodzi do miejsca, w którym wyświetlany jest popularny "Game over". Możemy wtedy rozpocząć nową grę bez ponownego wgrywania. W programie tym poprawiamy ostatni segment JETSET.SBF i poprawiony nagrywamy na taśmę w miejsce oryginalnego.

```
10 MEMORY 9999:LOAD "JETSET.SBF",10000
20 FOR x=31844 TO 31848:READ a:POKE x,a:NEXT x
30 DATA 205,30,187,0,194
40 SAVE "JETSET.SBF",6,10000,28168
```



Leszek Rudak

Piąty do brydża

Brydż jest, jak wiadomo, karcianą grą dla czterech osób połączonych w pary. Każda para usiłuje ograć przeciwników. Niewtajemniczonym podaję, że gra w brydża polega na zapowiedzeniu, ile lew (bitek lub wziątek) weźmiemy (o ile partner nie będzie przeszkadzał ...), a potem na wzięciu właśnie tylu lew. Przeciwnicy też mają swoją politykę: chcą wziąć jak najwięcej, najlepiej tyle, żeby nam zabrakło.

Jak widać, brydż to gra trudna, a do tego rozmawia się szyfrem, np. zamiast "wezmę 7 lew, jeżeli pozwolicie mi przebijać kierami wszystko inne" mówi się "jedno kier" i nic więcej!. Ponadto w czasie licytacji widzi się tylko swoje karty (chyba że uda się zapuścić żurawia...), a w czasie rozgrywki nie wolno konsultować się z nikim. Tylko "dziadek" wykłada swoje karty, a potem może narzekać.

O zasadach gry piszę celowo. Przyglądałem się programowi komputerowemu do gry w brydża. Program ten zastępuje aż trzech graczy, a czasem nawet czterech! (ciekawe, jaką korzyść ma komputer z gry w brydża sam ze sobą?).

Bywa tak, że mamy wielką ochotę zagrać i mamy nawet karty, ale okrzyki "CZWARTY !!!" nic nie dają. Często brakuje nawet drugiego i trzeciego... W takiej sytuacji komputer i program "Colossus Bridge 4" może przyjść z pomocą.

Z tym programem można pograć sobie w brydża na dostatecznym poziomie. Bez szczególnie błyskotliwych zagrań, bez brawurowych licytacji, ale bardzo rzetelnie. Program ten jest dobrą pomocą dla początkujących (zastrzeżenia patrz niżej), zwłaszcza tych, którzy wstydzą się grać nic nie umiejąc lub mają nerwowych partnerów. Niestety nie wydaje mi się, by dobry gracz mógł się czegoś nowego nauczyć. Raczej będzie musiał dostosować swój sposób licytacji do programu, w obronie zrezygnować z sygnalizacji (program nie zna Lavinthala).

Rasowy brydżysta może pokazać swój "lwi pazur" tylko w rozgrywce. Pozostaje jednak pytanie "po co ?" Program gra w obronie standardowo, bez zbytnich utrudnień, a wręcz przeciwnie pomaga jak może. Zdarzyło się nawet, że odrzucił blotkę od trzeciej damy mając do wyrzucenia bezwartościowe blotki w innym kolorze.

Napisałem, że można pograć sobie na dostatecznym poziomie. Powinienem w tym miejscu dodać, iż jest to poziom typowej gry "w kółku": na wczasach, na imieninach u Pani Ziuty itp. Zwykle z przypadkowym partnerem grywa się kiepsko. W przypadku programu "Colossus Bridge 4" jest trochę lepiej. Do dyskietki z programem dołączona jest niewielka broszurka, a w niej opis systemu licytacji, jakim posługują się wszyscy trzej komputerowi partnerzy. To jest dobra strona: wszyscy oni grają tak samo! Konsekwentnie realizują swój system licytacyjny, nawet wtedy gdy zdrowy rozsądek podpowiada coś

> 22

rys. Piotr Kakiet



21 <

innego. Z takim pewnym partnerem gra się znacznie lepiej niż z doskonałym graczem, który jednak nie przestrzega żadnych reguł. W "Colossus Bridge 4" na pewno nie będziemy grać 5 karo na trzech atutach.

Bardziej zaawansowanym brydżystom postaram się w skrócie przedstawić system licytacji zaimplementowany w programie "Colossus Bridge 4". Jak można się domyślić, system ten oparty jest na Acolu (program przecież pisali Anglosasi). Właściwie jest to mieszanina Acolu i licytacji naturalnej: słabe bez atu, silne dwa w kolor i blokujące trzy w kolor na siedmiokartowym longerze. Niestety zrezygnowano z bardzo wygodnego Acolowskiego otwarcia 2 trefle: odzywka ta pozostała tylko odzywką forsującą o sile od 22 PC. Natomiast program umie Staymana (pytanie o czwórki po otwarciu bez atu) i Blackwooda (pytanie o asy). Licytacja broniących jest też bardzo ograniczona. Z trudniejszych sztuczek stosuje tylko silną kontrę wywoławczą (z siłą co najmniej 13 PC).

Zasady *wistu* nie zostały opisane. Z kilku rozgrywek można domniemywać, że program *wistuje* naturalnie: z sekwensu – starszą, z blotek – najwyższą. Nie zauważyłem żadnej rozsądnej sygnalizacji *wistem* ani dołożeniem do lewy. Przypuszczam, że są to już zbyt trudne sztuczki dla komputera klasy IBM PC/XT.

Wracam do testu programu. Na początek strona graficzna. Ładne ramki i ozdobniki. Karty w rozgrywce prostokątne z symbolami kolorów, cyframi i literami oznaczającymi wysokości kart. Niestety całość rysuje się denerwująco wolno.

W czasie licytacji i rozgrywki dużo informacji na ekranie: kto rozdawał, jaki kontrakt, założenia rozdania. W przypadku licytacji mamy tylko ostatnie odzywki, a niestety brak całego przebiegu licytacji. W czasie rozgrywki stale jest wyświetlana poprzednia lewa. Niestety cały ekran sprawia wrażenie bałaganu. Do tego dochodzi jeszcze niedogodność w postaci wyświetlania kart w ręce tak jak są drukowane w książkach i czasopismach: w poziomych rzędach, najpierw symbol koloru, a za nim wszystkie wartości kart. Taki system jest do przyjęcia dla teoretyków lub zaawansowanych graczy, ale znacznie utrudnia orientację w grze osobie stawiającej pierwsze kroki w brydżu.

Obsługa programu jest bardzo łatwa. Licytuje się i dorzuca karty według tego samego systemu: najpierw cyfra lub nazwa karty, potem pierwsza litera nazwy koloru (oczywiście angielskiej nazwy). Naciśnięcie spacji w licytacji znaczy "PAS" w rozgrywce: zrób tak jak sam uważasz (czyli tzw. autoplay).

Program ma możliwość wyboru prędkości gry, ale na każdej z nich gra identycznie: popełnia te same błędy. Widocznie zmienia się tylko czas między kolejnymi odzywkami i *wistami*. Do przyjemnych opcji programu należą: możliwość wprowadzania rozdania z klawiatury i powtórzenia od początku dopiero co zakończonego rozdania. Można także na życzenie dostać dobre lub złe karty. Wystarczy podać, ile punktów Miltona chce się mieć na ręce.

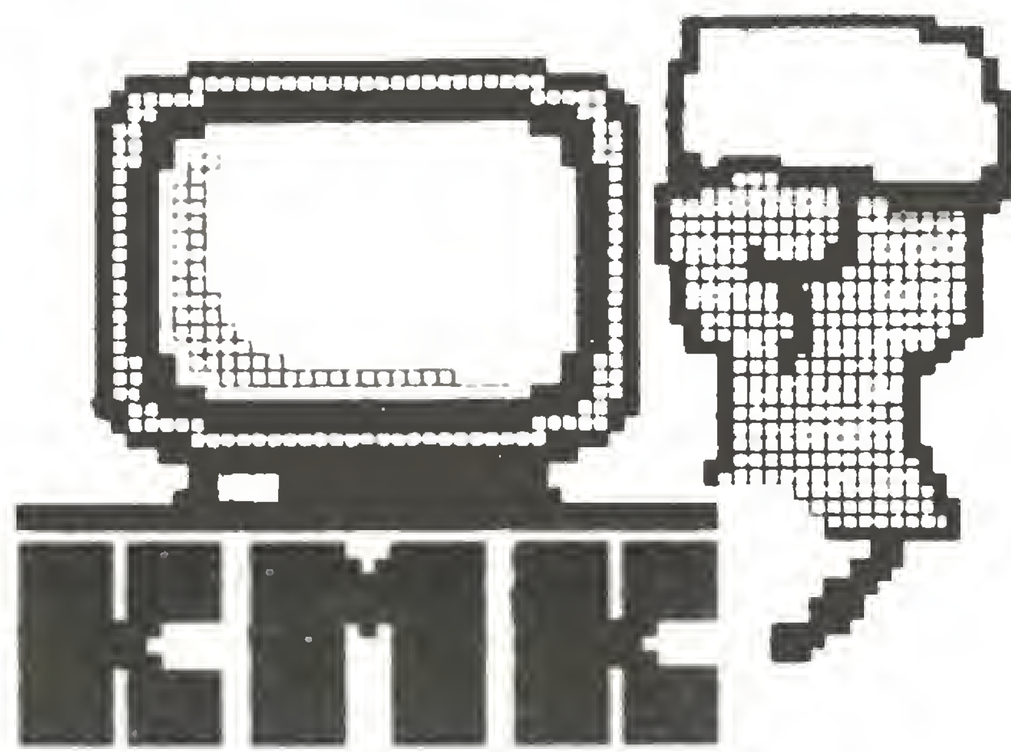
Niestety są też pewne niewygody. W czasie wprowadzania z klawiatury ułożonego przez siebie rozdania kontrola poprawności odłożona jest na koniec. Gdy już wszystko jest wprowadzone (to znaczy już podaliśmy wszystkie 52 karty), ale z błędem, dowiadujemy się, że np. "5 pik powtarza się" i wprowadzanie trzeba zaczynać od początku.

Program sam prowadzi zapis robrowy. Ponadto dba także o to, by grający się nie nudził. Zazwyczaj gracz siedzi na pozycji S. Gdy jednak para NS rozgrywa kontrakt z ręki N, stolik się obraca i my rozgrywamy. Wygląda to tak, jakby nasz partner nie miał zaufania do siebie i nam "wciskał" swoje karty z błaganiem w oczach: "Ty to rozegraj". Podobne "kółkowe" zwyczaje występują w rozgrywce, gdy ktoś przebija atutem. Program sygnalizuje taką "nietypową" sytuację klaksonem. W licytacji też powtarza się zabawna scena. Gdy któryś z graczy licytuje 4 bez atu program od razu, bez pytania stwierdza "Blackwood" żeby nikt nie miał wątpliwości.

Na zakończenie chcę przypomnieć, że program prawdopodobnie przeznaczony jest dla początkujących graczy. Utwierdzają mnie w tym próby z testami licytacji. Z różnych książek wybrałem kilka rozdań omówionych i przeanalizowanych przez mistrzów. Zleciłem programowi "Colossus Bridge 4" grę sam ze sobą. Niestety tylko w jednym przypadku program zagrał poprawnie (ale nie bardzo dobrze), w wielu pozostałych przypadkach program wymyślił zupełnie nowe kontrakty i nowe sposoby rozgrywki. Takie, jakich nie przewidzieli nawet mistrzowie.

W takiej sytuacji jedynym komplementem, który mogę powiedzieć jest to, że program nie zagląda w karty przeciwnikom. Gdyby bowiem podglądał, trudniej byłoby go ograć.

Testowany program "Colossus Bridge 4". Autorzy Hi-Tech Software. Program przeznaczony na komputery zgodne z IBM PC/XT z CGA (na komputerach z kartą Hercules trzeba używać emulatora karty CGA).



PĘTLICZEK – bo pętla jest podstawą programowania. Tu znajdziesz kolejną porcję zadań naszego Klubu Mistrzów Komputera. MĘTLICZEK – bo znajdziesz tu różne różności, związane z minikomputerem tak cienką nitką, że Redakcja już nie bierze za nią odpowiedzialności.

Redakcja strony klubowej: Marcin Jędrzejewski, Leszek Rudak.

DO CZYTELNIKÓW

Mamy za sobą kolejne, siódme już półrocze działalności Klubu Mistrzów Komputera. Czas na podsumowanie – na początku oczywiście lista członków KMK.

Lista członków KMK od lipca 1989
(lista ułożona 15 maja 1989 r.)

1.	Tadeusz Binek	KRAKÓW
2.	Bartłomiej Brzęczek	CZECHOWICE DZ.
3.	Jacek Cenzartowicz	SZCZECIN
4.	Bogdan Chmiela	ZIELONKI
5.	Andrzej Cieszyński	BYDGOSZCZ
6.	Piotr Fabian	GLIWICE
7.	Piotr Gzyl	NYSA
8.	Piotr Indyk	GDYNIA
9.	Mariusz Janus	POZNAŃ
10.	Wojciech Komnata	KRAKÓW
11.	Jerzy Labocha	SZCZECIN
12.	Katarzyna Lange	BELCHATÓW
13.	Maciej Łopaciński	ZAMOŚĆ
14.	Dariusz Mazur	ZABRZE
15.	Adam Nowicki	WROCLAW
16.	Andrzej Paszewin	WARSZAWA
17.	Piotr Ryszkiewicz	WODZISLAW ŚLĄSKI
18.	Andrzej Samojednik	HORYNIEC ZDRÓJ
19.	Zbigniew Sawczuk	BYDGOSZCZ
20.	Maciej Sawicki	WARSZAWA
21.	Jacek Stępień	BYTOM
22.	Krzysztof Syrek	PLOCK
23.	Arkadiusz Zacharzewski	GDAŃSK
24.	Tomasz Załęcki	MYŚLACHOWICE
25.	Jacek Zapała	RADOM
26.	Piotr Zawicki	WARSZAWA

W sprawdzaniu zadań pomagał członek KMK **Krzysztof Syrek**. Serdecznie dziękujemy. Pragnęlibyśmy jednocześnie przeprosić Pana Andrzeja Endlera ze Zduńskiej Woli za błędne wydrukowanie nazwiska w styczniowej liście członków KMK.

Napłynęło bardzo dużo rozwiązań, ale w dalszym ciągu otrzymujemy niewiele propozycji prostych, ciekawych programów, sztuczek itp. do zamieszczenia na naszej klubowej stronie. Okazało się, że najczęściej podejmowanym przez Czytelników (i zarazem krytykowanym) było zadanie 32/88 – zapis liczby dziesiętnej w systemie liczb Fibonacciego. Forma typowego rozwiązania praktycznie nie zmieniła się – jest to tekst gotowego programu z minimalnym komentarzem. Niestety, do rzadkości należy elegancki opis algorytmu rozwiązania. Oto statystyczne zestawienie języków programowania użytych w rozwiązaniach:

Basic	– 55 %
Pascal	– 25 %
C	– 11 %
Asembler	– 8 %
inne	– 1 %

Jak widać, w dalszym ciągu panuje Basic – i to z reguły w najgorszym, "rabunkowym" wydaniu. Programy wprawdzie działają poprawnie, ale pełno w nich magicznych POKE-ów, znienacka pojawiających się zmiennych o nic nie mówiących nazwach, skoków GOTO itd.

Bardzo cieszy mnie popularność Pascala i C. Niektórzy Czytelnicy nadsyłają programy w C, mimo że nie mogą ich uruchomić – widać moda na ten język trafiła też do KMK. Rozwiązania w tych językach są zazwyczaj czytelne i poprawne. Jest to wymuszone przez konieczność deklarowania struktury danych i blokową konstrukcję programów. Zdarzyło się jednak rozwiązanie w Pascalu z zupełnie nie uzasadnioną instrukcją skoku.

Rozwiązania w asemblerze są sformułowane przejrzyście. Właściwy tekst programu poprzedzony jest na ogół obszernym opisem metody, poszczególnym instrukcjom towarzyszą stosowne uwagi. Bez takiego podejścia napisanie programu w asemblerze byłoby niemożliwe. Wydaje mi się, że warto pisać rozwiązania właśnie w ten sposób. Opis metody powinien być krótki, ale wyczerpujący. Pewne informacje o tym, co rozwiązujemy, w jaki sposób, jakiego sprzętu i oprogramowania użyliśmy, należy umieścić na początku programu – bardzo ułatwią potem przeglądanie wydruków, zwłaszcza gdy upłynie trochę czasu... Komentowanie programu wielu z nas uważa za ekstrawagancję. Nie wnoszą nic konstruktywnego i obciążają zazwyczaj skromną pamięć komputera, a przecież autor i tak wie, co napisał, użytkownika zaś to nie interesuje. Zauważmy tylko, że duży program, który jest czytelny dziś, za tydzień jest już mniej czytelny, a za miesiąc prawie wcale. Dlatego dla własnej wygody opłaca się starannie dobierać nazwy dla zmiennych i podprogramów, zaznaczyć do czego one służą, umieścić odpowiednie nagłówki, które pomogą potem wychwycić cel cyklu operacji. Jeżeli piszemy poważny program w zespole, to umieszczanie komentarzy jest konieczne – bez nich nie będzie można łatwo połączyć poszczególnych fragmentów. Nie należy oczywiście przesadzać, komentarze nie mogą zaciemnić tekstu programu.

Praktycznie każda instrukcja

skoku czyni program zagmatwanym. Wyjątkiem jest tu instrukcja "break" w języku C, która powoduje wyjście z najbliższej pętli lub warunku. Właściciele małych komputerów bez możliwości pracy w tak wygodnych językach, jak Pascal czy C, powinni mimo wszystko redukować ilość skoków. O wiele lepiej jest posługiwać się podprogramami, definiować funkcje. Warto natomiast za pomocą instrukcji skoku skonstruować "pseudopętlę" WHILE-DO czy REPEAT-UNTIL, trzeba to tylko zaznaczyć w komentarzu. Oczywiście wewnątrz takiej pętli nie powinno się już używać skoków. Jeżeli warunek logiczny w pętli WHILE-DO jest spełniony, to wykonywany jest blok instrukcji, ponownie sprawdzany jest warunek itd. Jeżeli warunek ten nie jest spełniony, to oczywiście nic się nie stanie i program wykona polecenia znajdujące się za blokiem instrukcji. Oto przykład realizacji pętli:

```
130 REM pętla "while"; jeżeli
        warunek nie jest
        spełniony to
        przechodzimy do
        wiersza 280
140 IF NOT( warunek ) THEN
GOTO 280
150 REM początek bloku in-
strukcji
:
260 REM koniec bloku instruk-
cji
270 GOTO 140
280 ..
```

Pętla REPEAT-UNTIL różni się tylko tym, że najpierw wykonywany jest blok instrukcji, a dopiero potem sprawdzany warunek i jeżeli jest spełniony wychodzimy z pętli – tak więc blok ten wykonany będzie co najmniej raz:

```
130 REM pętla "repeat"; spraw-
        dzenie warunku w
        wierszu 270
140 REM początek bloku in-
        strukcji
:
260 REM koniec bloku instruk-
        cji – jeżeli warunek
        nie jest spełniony, to
        wracamy do 140
270 IF NOT( warunek ) THEN
GOTO 140
280 ..
```

M.J.

ZADANIA KLUBOWE

19/89. Układanka "Pentomino" to zestaw kilku klocków składających się z 5 kwadratów każdy. W zestawie "Pentomino" występują wszystkie możliwe połączenia 5 kwadracików, z pominięciem układów symetrycznych i "lustrzanych odbić".

Proponuję napisać program tworzący n-tomino, czyli układankę złożoną ze wszystkich klocków będących połączeniem n kwadratów (tak jak w "Pentomino"). Liczba kwadratów n powinna być daną dla programu.

L.R.

20/89. Gdy będziemy wypisywali kolejne liczby naturalne, jedna za drugą, to uzyskamy ciąg cyfr zaczynający się tak:

12345678910111213141516...

Proponuję napisać program, który dla danej liczby n poda, jaka cyfra stoi na miejscu n w powyższym ciągu.

L.R.

21/89. Proponuję napisać program pomagający w obsłudze nośników magnetycznych. Program ten powinien podać, ile miejsca potrzeba na nośniku magnetycznym do zapisania programu, który właśnie został wprowadzony do pamięci komputera. Wynik powinien być podany w postaci czasu nagrywania, gdy używamy magnetofonu lub w postaci liczby sektorów, gdy używamy dyskietek.

L.R.

O REKURENCJI [2]

Miesiąc temu w KMK kreśliliśmy za pomocą procedury rekurencyjnej prosty rysunek złożony z okręgów. Teraz proponuję prześledzić na prostych przykładach mechanizm wywoływania rekurencji.

Często w wydrukach dołączamy po kilka pustych wierszy – wydruk staje się wtedy jaśniejszy. Normalna procedura iteracyjna będzie wyglądać tak:

```
procedure linie ( k : integer );
  var
    j : integer;
  begin
    for j := 1 to k do
      writeln
    end;
```

Zagadnienie można rozwiązać też rekurencyjnie:

```
procedure linie2 ( k : integer );
  begin
    if k > 1 then linie (k – 1);
    writeln
  end;
```

Procedura iteracyjna nie wymaga komentarza. Sprawdźmy natomiast, co dzieje się z procedurą "linie2" wywołaną np. od parametru 3. Sprawdzony zostanie warunek – jest oczywiście spełniony, więc zostanie wywołana procedura "linie2" – od parametru 2. W praktyce komputer robi w pamięci kopię tej procedury i wywołuje właśnie tę kopię. Oczywiście zmienne kopii są zmiennymi lokalnymi ("widzianymi" tylko przez nią), więc żadne informacje nie są tracone. Warunek ponownie jest spełniony, więc znowu jest wywołana procedura "linie2" – od parametru 1. Zauważmy, że do tej pory nie wykonała się żadna instrukcja "writeln". Tym razem warunek nie jest spełniony – komputer drukuje pusty wiersz i wraca do tego miejsca w poprzedniej kopii, z którego została wywołana procedura "linia2". Teraz dopiero drukowany jest "zaległy" wiersz z tej kopii procedury i znowu następuje powrót – już do macierzystej procedury. Co by się stało, gdybyśmy zamienili kolejność poleceń w naszej procedurze? Najpierw drukowany by był pusty wiersz, a dopiero potem tworzone (na omówionych już zasadach) kolejne kopie.

Za pomocą procedur rekurencyjnych wyjątkowo wygodnie obsługuje się listy. Przypuśćmy, że dysponujemy listą prostą zbudowaną w oparciu o następujące deklaracje (są one dobrze znane czytelnikom cyklu "Wskaźniki w Pascalu"):

```
type
  wskaźnik = ^kostka;
  kostka = record
    wartość : integer;
    strzałka : wskaźnik
  end;
```

Oto procedura drukująca taką listę:

```
procedure drukowanie ( var
początek : wskaźnik );
  begin
    if not( początek = nil ) then
      begin
        writeln( początek^.
wartość );
        drukowanie( początek
^.strzałka )
      end
  end;
```

Procedura drukuje wartość pierwszego elementu listy, przeskakuje na kolejny i powtarza powyższą czynność. Warto zauważyć, że jeżeli tu zamienimy kolejność poleceń w bloku warunku otrzymamy procedurę, która drukuje listę w odwrotnej kolejności:

```
procedure drukowanie2 ( var
początek : wskaźnik );
  begin
    if not( początek = nil ) then
      begin
        drukowanie2 ( począ-
tek^.strzałka );
        writeln(       począ-
tek^.wartość )
      end
  end;
```

M.J.

POSTULATY EULERA

1. Nie należy zapisywać obliczeniami sześćdziesięciu stron po to, aby uzyskać wynik, który można osiągnąć w dziesięciu wierszach po pewnych uważnych przemyśleniach.
2. Jeżeli chcemy spędzić trzy dni i trzy noce z rzędu na obliczeniach, to warto wcześniej poświęcić kwadrans, aby stwierdzić, które reguły obliczeń będą najbardziej przydatne.
3. Jeżeli wynik starannych obliczeń wydaje się być w niezgodzie z rzeczywistością, to musimy ufać naszym obliczeniom bardziej niż naszym zmysłom.



Drodzy Czytelnicy!

"Forum" to rubryka przeznaczona w całości do Waszej dyspozycji. Możecie do niej pisać nie tylko o swoich osiągnięciach, nadsyłać użyteczne programiki czy "sztuczki i chwyty", które odkryliście, ale możecie pisać także o problemach, które spotykacie w pracy z mikrokomputerami. Być może ktoś inny je rozwiązał i będzie mógł Wam tą drogą pomóc.

Dzisiaj prezentujemy: Wczytywanie rysunków do programów pisanych w GFA Basicu, procedurę wprowadzania danych w Turbo Pascalu, usprawnienie wprowadzania liczb i realizację przerwań dla ZX Spectrum, program obliczający parametry anteny TV typu Yagi dla Master Compact, procedurę ustalającą kolory tła, ramki, kursora dla Commodore 64, zegar wyświetlający na ekranie aktualny czas dla użytkowników C64, ciekawe podejście do problemu animacji komputerowej (IBM PC) oraz użyteczny programik wygaszający po pewnym czasie ekran w komputerze ZX Spectrum, dzielenie z dużą dokładnością (Atari XL/XE) oraz ustawienie skosu głowicy (Commodore C16).

Zapraszamy

Wczytywanie rysunków do programów pisanych w GFA Basicu (Atari ST)

W pewnych przypadkach zachodzi potrzeba wykonania skomplikowanego rysunku, który zostanie użyty we własnym programie (np. jako ilustracja czy plansza informacyjna). Czasami łatwiej jest wykonać rysunek za pomocą programu graficznego niż bezpośrednio w Basicu i wczytywać go na ekran w trakcie wykonywania programu. Ze względu na dużą liczbę elementów oraz trudne do odwzorowania w Basicu krzywe rysunek przykładowy został wykonany za pomocą programu DEGAS Elite (na monitorze monochromatycznym w trybie wysokiej rozdzielczości) i zapamiętany na dysku bez kondensacji (nazwa z rozszerzeniem ".PI3"). Zajmuje on 32066 bajty, z czego rysunek mieści się od 34 do 32034 bajtu. Wczytanie rysunku podczas wykonywania programu odbywa się dzięki następującej procedurze:

```
PROCEDURE Czytanie_rysunku
  OPEN "I",#1,"NAZWA.PI3"
!Otwieranie zbioru z rysunkiem
  SEEK #1,34 !Przesuwanie gło-
wicy na 34 bajt
  BGET #1,XBIOS(2),32000
!Wczytywanie rysunku na ekran
  CLOSE #1
  RETURN
```

> 24

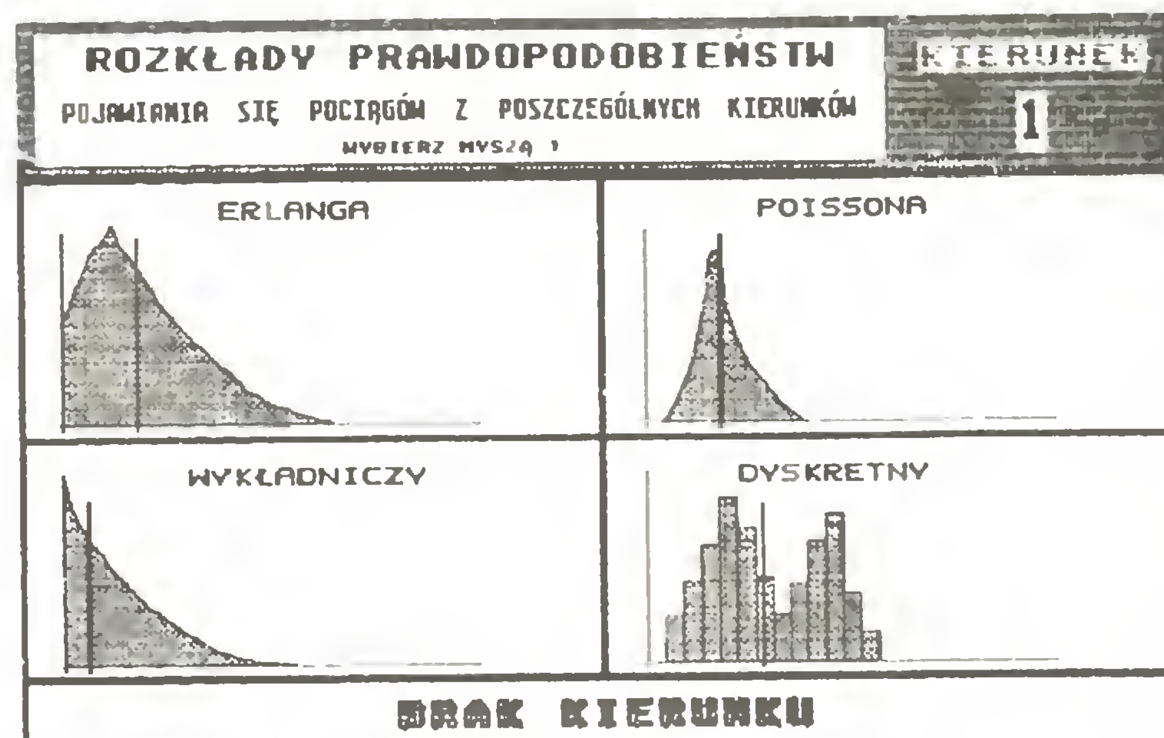
23 <

Przedstawiony sposób ma jednak tę wadę, że wczytanie rysunku trwa ponad 2,5 sek. (a razem z otwarciem zbioru – 4,6 sek.) oraz przy każdorazowym wczytaniu komputer kręci dyskietką. Aby tego uniknąć, można rysunek wczytać do zmiennej tekstowej, np. w czasie, gdy użytkownik czyta informacje z ekranu (w moim programie podczas wyświetlania informacji na temat wykonywanego programu). Należy wtedy zmienić wiersz BGET ... na:

RY$=INPUT$(32000,#1) !RY$ – nazwa zmiennej z rysunkiem

Umieszczanie zapamiętanego rysunku na ekranie następuje przy użyciu instrukcji SPUT RYS$, co trwa 0,2 sek. i ma tę zaletę, że można to robić wielokrotnie.

Tomasz Kurzacz
Klub Atari ST
Warszawa



Rysunek wczytywany z dysku w trakcie wykonywania programu

Wygodne wprowadzanie liczb (ZX Spectrum)

Do wprowadzania danych liczbowych z klawiatury stosuje się zazwyczaj instrukcję INPUT. Jest to nieco uciążliwe, szczególnie przy częstym występowaniu ułamków dziesiętnych. Wymaga bowiem każdorazowego przytrzymania klawisza SYMBOL SHIFT przy naciskaniu klawisza kropki dziesiętnej. Mylne wciśnięcie klawisza literowego i nieopatrzne przesłanie tej informacji do systemu powoduje przerwanie realizacji programu.

Zamieszczony podprogram jest pozbawiony tych wad. Zamiast **INPUT a** wystarczy zastosować **GOSUB wygodne wprowadzanie liczb LET a**=wynik.

Zaletą podprogramu jest możliwość wprowadzania kropki dziesiętnej bez naciskania klawisza SYMBOL SHIFT. Wiersze 8570 i 8660 (analogicznie wiersze 8720 i 8810) są potrzebne, bowiem ich brak może powodować błędy przy szybkim wciskaniu klawiszy (przyjęcie poprzedniej cyfry, mimo wciśnięcia innego klawisza).

Chcąc wprowadzić zero wystarczy wcisnąć tylko klawisz ENTER. Podprogram przyjmuje daną cyfrę jednokrotnie, niezależnie od czasu naciskania klawisza. Naciśnięcie DELETE usuwa całą wprowadzaną liczbę.

Marek Nowosad
Lublin

```
  10 LET wygodne wprowadanie liczb = 8500
  20 LET czesc druga = 8700
  30 LET wyjscie = 8850

8500 REM wygodne wprowadzanie liczb
8510 PRINT AT 21,0;"
8520 LET czesc calkowita = 0
8530 LET czesc ulamkowa = 0
8540 LET mnoznik = 1
8550 LET kolumna = 2
8560 PAUSE 0: IF INKEY$<>"m" AND INKEY$<>"M" AND INKEY$<>"." AND CODE
     INKEY$<>13 AND (CODE INKEY$<48 OR CODE INKEY$>57) AND CODE INKEY$<>12
     THEN GO TO 8560
8570 IF CODE INKEY$<>PEEK 23559 THEN GO TO 8560: REM czekanie az osmy bajt
     zmiennej KSTATE przyjmie kod aktualnie wciskanego klawisza
8580 IF PEEK 23559=12 THEN BEEP .5,-10: GO TO wygodne wprowadzanie liczb:
     REM nacisniecie DELETE powoduje wprowadzanie tej liczby od poczatku
8590 BEEP .05,40
8600 IF PEEK 23559=13 THEN GO TO wyjscie: REM w-ek jest spelniony, gdy
     ostatnio wciskanym klawiszem bylo ENTER
8610 IF PEEK 23559=46 OR PEEK 23559=77 OR PEEK 23559=109 THEN PRINT AT 21,
     kolumna;".";: LET kolumna = kolumna + 1: GO TO czesc druga: REM W-ek
     jest spelniony, gdy ostatnio wciskanym klawiszem bylo M lub m lub .
8620 LET ostatnia cyfra = VAL CHR$ PEEK 23559
8630 PRINT AT 21,kolumna;ostatnia cyfra;
8640 LET czesc calkowita = 10 * czesc calkowita + ostatnia cyfra
8650 LET kolumna = kolumna + 1
8660 IF CODE INKEY$=PEEK 23559 THEN GO TO 8660: REM czekanie na puszczenie
     klawisza
8670 GO TO 8560
8680
8700 REM czesc druga
8710 PAUSE 0: IF CODE INKEY$<>13 AND (CODE INKEY$<48 OR CODE INKEY$>57) AND
     CODE INKEY$<>12 THEN GO TO 8710
8720 IF CODE INKEY$<>PEEK 23559 THEN GO TO 8710: REM czekanie az osmy bajt
     zmiennej KSTATE przyjmie kod aktualnie wciskanego klawisza
8730 IF PEEK 23559=12 THEN BEEP .5,-10: GO TO wygodne wprowadzanie liczb:
     REM nacisniecie DELETE powoduje wprowadzanie tej liczby od poczatku
8740 BEEP .05,40
8750 IF PEEK 23559=13 THEN GO TO wyjscie: REM warunek - wcisniecie ENTER
8760 LET ostatnia cyfra = VAL CHR$ PEEK 23559
8770 PRINT AT 21,kolumna;ostatnia cyfra;
8780 LET kolumna = kolumna + 1
8790 LET mnoznik = mnoznik * .1
8800 LET czesc ulamkowa = czesc ulamkowa + mnoznik * ostatnia cyfra
8810 IF CODE INKEY$=PEEK 23559 THEN GO TO 8810: REM czekanie na puszczenie
     klawisza
8820 GO TO 8710
8830
8850 REM wyjscie z podprogramu
8860 LET wynik = czesc calkowita + czesc ulamkowa
8870 BEEP .1,12: BEEP .1,18
8880 RETURN
```

Procedura odczytu danych (Turbo Pascal)

W Turbo Pascalu brak jest procedur odczytu danych pozwalających na stałą kontrolę pobieranych z klawiatury znaków i ich odrzucenia lub akceptacji w zależności od narzuconego przez programistę formatu. Poniżej przedstawiam moją funkcję odczytu z klawiatury długiej liczby całkowitej (typ *longint*).

Za pomocą tego podprogramu możemy określić, z ilu cyfr może składać się liczba i w jakim miejscu ekranu ma być wyświetlana. Przy odczycie z klawiatury akceptowane są jedynie cyfry i znak minus (dla liczb które mają być ujemne). Przy wywołaniu funkcji należy przesłać liczbę, która jest jej wartością początkową i którą można modyfikować. Jeśli chcemy pobrać nową liczbę, należy przesłać 0. Odczyt może być przerwany w dowolnej chwili przez naciśnięcie klawisza ESC i wówczas do modułu wywołującego zostanie zwrócona liczba, która była wysłana do funkcji jako wartość początkowa. Jeżeli wartość odczytanej liczby jest spoza zakresu liczb typu *longint*, zostanie zwrócona wartość 0. W przeciwnym przypadku odesłana zostanie liczba o wartości odczytanej z klawiatury.

Włodzimierz Pszczółkowski

```
>>  Funkcja POBINT  <<

(***********************************************************************
 *  FUNKCJA PobInt -> pobranie z klawiatury liczby calkowitej          *
 *                    typu LONGINT                                     *
 *  PARAMETRY      -> WspX.WspY - wspolrzedne poczatku pola odczytu    *
 *                                liczby                               *
 *                 -> IleC      - liczba cyfr w odczytywanej liczbie   *
 *                                (maksymalnie do 11 znakow)           *
 *                 -> Dana      - wartosc poczatkowa liczby ktora mozna *
 *                                modyfikowac. jezeli nie mamy liczby  *
 *                                poczatkowej przeslij 0               *
 *                                                                     *
 *  WARTOSCI       -> zwraca pobrana liczbe jezeli zawiera sie ona w   *
 *                    zakresie (MININT.MAXINT). w przeciwnym przypadku *
 *                    zwraca 0.                                        *
 ***********************************************************************)

function pobint(wspx.wspy.ilec : byte: var dana : longint) : longint:
const
  cyfry = ['0'.'1'.'2'.'3'.'4'.'5'.'6'.'7'.'8'.'9']:
  minus = '-':
  maxint : longint =  2147483647:
  minint : longint = -2147483647:
  ESC=27 : ENTER=13:  DEL=83:

var
  max    : byte:                  (* maksymalna liczba cyfr w polu   *)
  ostx   : byte:                  (* wpsolrzedna ostatniej cyfry     *)
  ile    : byte:                  (* licznik liczb znaczacych        *)
  dane   : array[1..11] of char:  (* tablica wczytanych znakow       *)
  lan    : string[11]:            (* liczba wynikowa jako lancuch    *)
  wynikr : real:                  (* wynik jako liczba rzeczywista   *)
  wynik  : longint:               (* wynik jako liczba longint       *)
  i.j    : byte:
  pom    : integer:
  kod    : byte:                  (* kod ASCII wczytany z klawiatury *)


begin

 (* ==) ustalenie wartosci zmiennych *)
       max  :=ilec:
       ostx :=wspx+max-1:
       for i:=1 to max do dane[i]:=' ':

 (* ==) pole wpisowe *)
       textcolor(black):
       textbackground(white):
       gotoxy(wspx.wspy):
       for i:=1 to max do write(' '):

 (* ==) zamiana liczby na lancuch i wpisanie go do tablicy *)
       str(dana.lan):
       ile:=length(lan):
       j:=max:
       for i:=ile downto 1 do begin
           dane[j]:=lan[i]:
           dec(j)
       end:
       gotoxy(wspx.wspy):
       for i:=1 to max do write(dane[i]):
 >>  Funkcja POBINT  <<

 (* ==) odczyt klawiatury i wykonanie odpowiednich operacji *)
       gotoxy(ostx.wspy):
```

```
        repeat
           while keypressed do kod:=ord(readkey);
           kod:=ord(readkey);
           if kod=0 then kod:=ord(readkey);

           if (chr(kod) in CYFRY) or (chr(kod)=MINUS) then begin
              if chr(kod)=MINUS then
                 if (ile=1) and (dane[max]='0') then dane[max]:='-';
              if chr(kod) in CYFRY then
                 if ile<max then begin
                                  inc(ile);
                                  for i:=1 to (max-1) do dane[i]:=dane[i+1]
                                  dane[max]:=chr(kod)
                                end
                            else write(#7);
              gotoxy(wspx,wspy);
              i:=1;
              while (dane[i]=' ') or (dane[i]='0') do begin
                    if dane[i]='0' then begin
                       dane[i]:=' ';
                       dec(ile)
                    end;
                    inc(i)
              end;
              for i:=1 to max do write(dane[i]);
              gotoxy(ostx,wspy)
           end;

           if kod=DEL then begin
              if ile=1 then dane[max]:='0'
                       else begin
                              dec(ile);
                              for i:=max downto 1 do dane[i]:=dane[i-1];
                              dane[1]:=' '
                            end;
              gotoxy(wspx,wspy);
              for i:=1 to max do write(dane[i]);
              gotoxy(ostx,wspy)
           end
        until (kod=ENTER) or (kod=ESC);

(* ==) ustalenie wyniku koncowego *)
        if kod=ESC then wynik:=dana
                   else begin
                          lan:='';
                          for i:=1 to max do
                               if (dane[i] in CYFRY) or (dane[i]=MINUS) then
                                  lan:=lan+dane[i];
                          val(lan,wynikr,pom);
                          if (wynikr)=minint) and (wynikr(=maxint) then
                                                    wynik:=round(wynikr)
                                              else   wynik:=0;
                        end;
        textcolor(white);
        textbackground(black);
        pobint:=wynik

end;  (* KONIEC -) pobint *)
```

```
      >>  Program demonstracyjny  <<

      uses crt;
      var wzor,wynik : longint;


      (***********************************
       *                                 *
       *      Miejsce na procedure       *
       *            POBINT               *
       *                                 *
       ***********************************)


      begin
        textcolor(white);
        textbackground(black);
        clrscr;
        wzor:=123;
        wynik:=pobint(5,6,11,wzor);
        gotoxy(5,7):write(wynik);
      end.
```

AutoOnOff (ZX Spectrum)

Przedstawiam Wam własny program, ktory nadzoruje pracę ZX Spectrum w ten sposób, że po upływie określonego czasu (wstępnie 60 sek.) od ostatniego naciśnięcia dowolnego klawisza wygasza ekran telewizyjny przez wypełnienie obszaru pamięci atrybutów wartością 0 i BORDER 0. Bieg programu w Basicu zostaje zatrzymany, a po wciśnięciu dowolnego klawisza zawartość ekranu zostaje zrekonstruowana, a program w Basicu kontynuowany.

Program zajmuje ostatnie 300 bajtów pamięci, jest nierelokowalny, podczas wygaszania wykorzystuje bufor drukarki.

Niedopuszczalne jest podnoszenie ustalonego RAMTOP-u ponad 65235 oraz zmiana zawartości rejestru I.

- Uruchomienie – USR 65528.
- Wyłączenie – USR 65521.
- Stała czasowa (w sekundach) * 50 w komórkach 65519 i 65520.

Przy wyłączonych przerwaniach (LOAD, SAVE, VERIFY, BEEP, CIRCLE ...) odliczanie czasu jest zatrzymane.

W moim programie wykorzystałem programy:

- Rolanda Wacławka – Loader kodu,
- Piotra Zawickiego – HeToBa,
- Jerzego Ziaji – Kompresja ekranu,
- Andrzeja Grossmana – Znacznik.

Po wprowadzeniu programu z klawiatury wykonujemy GO TO 1000 i zapisujemy go na taśmie. Po zapisaniu po ok. 20 sek. zostanie wykonane NEW i teraz możemy zapisać na taśmie gotowy produkt – trzystubajtowy blok kodu: SAVE "AutoOnOffCODE"CODE65236,300.

Mariusz Nahajowski
Legnica

```
   1 DATA "65236,10"
   2 DATA "F5E5C5D5CD8E027B17D2C9FF2AEFFF7DB428092B22EFFFD1C1C33A003E5B32F703"
   3 DATA "FF21005BAF4777231OFCE51100401A4F21005B093AF7FFBE38013413BA20EFE153"
   4 DATA "1D3B56BA38027A4B1D10F67121005B11FF5B01011BEDB821FF40110040 3E5C3211"
   5 DATA "F7FFCD5CFF3AFF5BB9280904052805 3E0290300D3AFF5B121378121379121318EB"
   6 DATA "E179121310FC18DA3AF7FFBC2070E1D5210058360011015801FF02EDB036C7AF13"
   7 DATA "D3FECD8E027B1738F93A485C1F1F1FD3FEE1010040373FED423B3BE511FF5BEB75"
   8 DATA "ED52EBC1E12BD511FE5BEDB8210040D13E5B32F7FF1A4F3AFF5BB92010131A4728"
   9 DATA "131A1377CDE0FF380C10F818E81A1377CDE0FF30E021B80B22EFFFC3EBFE0601B6"
  10 DATA "7E4FCDE0FFD87EB9C00420F623083AF7FFBC20030837C908373FC9B80BED56C9BA"
  11 DATA "C3D4FE5BF33E3BED47ED5E18F3"
1000 SAVE "AutoOnOff" LINE 9993
9993 CLEAR 65235: READ A,S: FOR F-1 TO S
9994 READ L$: LET L-LENL$: LET S-0: LET K-2
9995 LET A$-L$(K-1): LET B$-L$(K)
9996 LET C-(CODE A$-48-(7*(A$>"@")))*16 + CODE B$-48-(7*(B$>"@"))
9997 IF K<L THEN POKE A,C: LET S-S+C: LET K-K+2: LET A-A+1: GO TO 9995
9998 IF S-256*INT(S/256)<>C THEN PRINT "BLAD W LINII ";F+1: STOP
9999 NEXT F: NEW
```

Zegar (Commodore 64)

Szanowna Redakcjo!

Przesyłam napisany przeze mnie program ZEGAR. Wyświetla on w dolnej ramce ekranu czas w formacie HH:MM:SS (godziny, minuty, sekundy). Program działa na Commodore 64 wykorzystując przerwania VIC. Pozwala na równoczesną pracę w Basicu i umożliwia obsługę magnetofonu (pod nadzorem TURBO) oraz wszystkich innych urządzeń peryferyjnych.

Załączony wydruk przedstawia program, który należy przepisać i uruchomić. W przypadku korzystania z magnetofonu w wierszu 240. liczbę 8 należy zmienić na 1. Ewentualne błędy w przepisywaniu zostaną zasygnalizowane komunikatem: ERROR IN [numer wiersza z błędem]. Jeśli nie będzie żadnych usterek, pojawi się pytanie "SAVE ?". Należy wtedy włożyć dysk (kasetę) i nacisnąć "Y". Właściwy program zostanie zapisany i będzie już gotowy do użytku. Wczytuje się go przez LOAD"ZEGAR",8,1 z dysku, a z kasety przez LOAD"ZEGAR",1,1 i uruchamia komendą: SYS 49152,HHMMSS, HHMMSS. Pierwszy parametr to czas aktualny, drugi – czas alarmu. Jeżeli nie chcemy ustawiać alarmu, możemy drugi parametr pominąć.

Wyświetlanie czasu można przerwać naciskając jednocześnie klawisze STOP i RESTORE.

Z poważaniem
Marcin Nowak
Kraków

```
10 REM ***** MARCIN  NOWAK *****
20 REM  ****** Z E G A R *****
30 REM          ** C64 **
100 S=49152
110 FOR F=0 TO 24:CN=0
120 FOR G=1 TO 18:READ A$
130 H=ASC(LEFT$(A$,1))-48
135 L=ASC(RIGHT$(A$,1))-48
140 IF H>9 THEN H=H-7
150 IF L>9 THEN L=L-7
160 P=H*16+L:CN=CN+P:POKE(S),P:S=S+1
170 NEXT:READ X
180 IFCN<>XTHENPRINT"ERROR IN";1000+2*F:END
190 NEXT
200 PRINT"SAVE ?"
210 GETK$:IF K$<>"T"THEN210
220 POKE 43,0:POKE 44,192:POKE 45,194
230 POKE 46,193:POKE 56,208
240 SAVE"ZEGAR",8:REM TU ZMIENIAC
1000 DATA A2,00,86,FB,AD,0E,DC,09,80,8D
1010 DATA 0E,DC,20,79,00,C9,2C,D0,2072
1020 DATA 15,20,8D,C1,CE,0D,C7,C9,2C,D0
1030 DATA 0B,A9,80,8D,0F,DC,20,8D,2115
1040 DATA C1,8E,0F,DC,8E,08,DC,8A,9D,00
1050 DATA C4,CA,D0,FA,8E,FF,FF,8E,2885
1060 DATA 20,D0,8E,21,D0,A9,13,85,02,A2
```

> 26

```
1070 DATA 03,BD,6D,C1,A8,A9,10,99,2108
1080 DATA 00,C4,A5,02,9D,F8,C7,C6,02,CA
1090 DATA 10,ED,78,A9,C0,8D,15,03,2268
1100 DATA A9,70,8D,14,03,A9,FA,8D,12,D0
1110 DATA A2,01,8E,1A,D0,A9,7F,8D,2207
1120 DATA 0D,DC,58,60,A9,FA,CD,12,D0,F0
1130 DATA 03,4C,7E,EA,AD,11,D0,A2,2506
1140 DATA 13,8E,11,D0,29,7B,09,0B,48,AD
1150 DATA 00,DD,48,29,FC,8D,00,DD,1763
1160 DATA A2,00,BD,71,C1,A8,B9,00,D0,99
1170 DATA 00,C8,BD,7F,C1,99,00,D0,2441
1180 DATA E8,96,D0,EC,A5,FE,F0,2C,30,05
1190 DATA A9,0C,8D,15,D0,C6,FB,C6,2767
1200 DATA FB,C6,FB,D0,1D,C6,FB,8D,00,D4
1210 DATA 8D,04,D4,8D,06,D4,A9,0F,2639
1220 DATA 8D,18,D4,A9,0C,8D,01,D4,8D,05
1230 DATA D4,A9,11,8D,04,D4,A2,02,1977
1240 DATA AD,0B,DC,C9,12,D0,03,A9,00,18
1250 DATA 90,0F,29,7F,C9,12,F0,09,1822
1260 DATA F8,69,12,D8,90,03,FD,09,DC,48
1270 DATA 38,FD,0B,C7,95,FC,68,48,2320
1280 DATA 4A,4A,4A,4A,99,00,C7,C8,68,29
1290 DATA 0F,99,00,C7,C8,CA,10,E2,2004
1300 DATA AD,08,DC,A5,01,48,29,FB,85,01
1310 DATA 88,B9,00,C7,0A,0A,0A,AA,1785
1320 DATA E8,84,02,B9,67,C1,A8,BD,7F,D1
1330 DATA 99,00,C4,C8,C8,C8,E8,8A,2859
1340 DATA 29,07,D0,F1,A4,02,D0,DE,68,85
1350 DATA 01,A5,FC,05,FD,05,FE,D0,2473
1360 DATA 02,C6,FB,A2,01,8E,19,D0,CA,BD
1370 DATA 71,C1,A8,B9,00,C8,99,00,2392
1380 DATA D0,E8,E0,0E,D0,F1,68,8D,00,DD
1390 DATA 68,8D,11,D0,4C,31,EA,00,2422
1400 DATA 01,40,41,80,81,0B,11,4B,51,01
1410 DATA 02,03,04,05,10,15,17,1C,674
1420 DATA 1D,27,28,29,00,0A,50,0A,80,0A
1430 DATA 00,07,00,00,07,07,07,07,422
1440 DATA 20,A2,03,20,73,00,B0,2B,E9,2F
1450 DATA 0A,0A,0A,0A,85,02,20,73,1165
1460 DATA 00,B0,1E,E9,2F,05,02,9D,08,DC
1470 DATA 9D,0A,C7,CA,D0,E1,AD,0D,2065
1480 DATA C7,C9,13,90,04,F8,E9,32,D8,8D
1490 DATA 0B,DC,4C,73,00,4C,99,AD,2279
```

Animacja komputerowa (IBM PC)

Szanowna Redakcjo!

Wiele osób wykorzystuje komputer do tworzenia grafiki oraz animacji komputerowych.

Problem stworzenia iluzji ruchu często komplikuje postawione zadanie. Często wystarcza przysłowiowy "rzut oka" na podobny problem, aby uporać się z własnym zadaniem. Zasada wywoływania złudzenia ruchu polega na przemiennym rysowaniu na stronie aktywnej, natomiast wyświetlaniu na ekranie monitora strony wcześniej narysowanej. Powoduje to, że proces rysowania jest niewidoczny, a iluzja zależy jedynie od szybkości przełączania obu stron. Przedstawiony poniżej program opiera się właśnie na tej zasadzie. Oczywiście nie rozwiązuje to problemu animacji komputerowej, a jedynie sygnalizuje podejście.

Program został napisany w języku Microsoft C ver. 5.0, lecz bez żadnych zmian może być również kompilowany w Microsoft Quick C ver. 1.0.

Łączę wyrazy szacunku
Włodzimierz Mozol
Szczecin

```
#include <conio.h>
#include <graph.h>
#include <math.h>
#include <stdlib.h>

/* Demonstration program for EGA, (c) 1989 by Wlodzimierz Mozol */

#define PI 3.14159265358        /* Constant Pi */
#define XOFFS 320               /* Coordinate X */
#define YOFFS 250               /* Coordinate Y */
#define HI 150                  /* High */
#define AEL 150                 /* A - axis of elypse */
#define BEL 60                  /* B - axis of elypse */
#define NEWX 100                /* X - axis 3 - rd point */
#define NEWY 20                 /* Y - axis 3 - rd point */

void main()
{
int p1x,p1y,p2x,p2y,p3x,p3y,page=2;
double fi;
struct videoconfig config;
fi = PI / 16;
 getvideoconfig(&config);
if(config.adapter != EGA) {
        (printf("EGA not available,program terminated. n");
        exit(1);
}
switch config.memory) {
        case 64 :  setvideomode( ERESNOCOLOR);
                   break;
        default :  setvideomode( ERESCOLOR);
                   setcolor(2);
                   break;
.
while  !(kbhit()) ) {
        fi = (fi  ( 2 * PI)) ? (PI / 8) : (fi + PI / 16);
        p1x = (int)(XOFFS + AEL * cos(fi));
        p1y = (int)(YOFFS + BEL * sin(fi));
        p2x = (int)(XOFFS - AEL * cos(fi));
        p2y = (int)(YOFFS - BEL * sin(fi));
        p3x = p1x + NEWX;
        p3y = p1y + NEWY;
         moveto(p1x,p1y);
         lineto(p2x,p2y);
         lineto(XOFFS,YOFFS - HI);
         lineto(p1x,p1y);
         lineto(p3x,p3y);
         lineto(p2x,p2y);
         moveto(p3x,p3y);
         lineto(XOFFS,YOFFS - HI);
        page = (page % 2) ? 2 : 3;
         setvisualpage(!(page - 2));
         setactivepage(page - 2);
         clearscreen( GCLEARSCREEN);
}
 setvideomode( TEXTC80);
```

Obsługa przerwań (ZX Spectrum)

Redakcjo "Komputera"!

Mikrokomputer ZX Spectrum ma bardzo niewygodną obsługę przerwań niemaskowalnych (przerwania maskowalne wskutek błędu w pamięci ROM są niemożliwe). Otóż chcąc stosować przerwania musimy wypełnić 257 bajtów w wybranej przez programistę stronie pamięci. Mój program wykorzystuje wolną (zapełnioną liczbami 255), 58. stronę pamięci Spectrum.

10 REM Wygodne przerwania
20 REM M. Lipczynski 1II1989
30 CLEAR 65336: FOR i=65505 TO 65535: READ j: POKE i,j: NEXT i: POKE 23675,57: STOP
40 DATA 237,86,201,62,58,237, 71,237,94,201
50 DATA 197,213,205,0,0,209, 193,24,70,245
60 DATA 229,33,254,255,53,40, 239,24,60,24,24

Teraz przerwania włącza się za pomocą komendy USR 65508, a wyłącza USR 65505. Adres procedury przerwań znajduje się w komórkach 65518 i 65519 (początkowo są tam zera, więc trzeba wpisać adres, zanim włączy się przerwania), zaś licznik (0,02 s pozostałe do przerwania) w komórce 65534. Dodatkowo program obniża wartość UDG. W procedurze użytkownika można zastosować jedynie rejestry AF, BC, DE, i HL (jeśli chce się używać innych, to procedura musi je sama przechować). Oto przykład takiej procedury (należy ją dołączyć do podanego programu):

70 DATA 54,50,62,7,211,254,201
80 FOR i=23296 TO 23302: READ j: POKE i,j: NEXT j: POKE 65519,91
90 RANDOMIZE USR 65508: BORDER 6

Procedura pozwala również uzyskać kolor ramki inny niż okna informacyjnego (ramka robi się biała co 1 s, a okno systemowe jest żółte).

Mateusz Lipczyński (12 lat)
Sopot

Obliczanie parametrów anteny TV (Master Compact)

Szanowna Redakcjo!

Od roku mam komputer Master Compact i bardzo ubolewam, że w żadnym czasopiśmie o tematyce komputerowej nie mogę nic dla niego znaleźć. Aby pocieszyć wielu takich jak ja i w minimalnym stopniu przyczynić się do wypełniania tej luki, postanowiłem przesłać Wam mój program obliczający parametry anteny TV typu Yagi.

Może ktoś zechce zamieścić na Waszych łamach program do przemieszczania obiektów po ekranie za pomocą joysticka, tzn. coś na wzór atarowskiej techniki P-M.

Janusz Gawlik
Sosnowiec

1 MO.129:DIM d(11),l(11)
2 P.C"Jest to program na obliczanie parametrow anteny TV typu Yagi."CC"Jezeli chcesz obliczyc wymiary anteny na ktorys z kanalow stacji zachodnich, to jego numer podaj jako setna czesc calosci."
3 P.C"Dla UKF-u zas za numer kanalu wpisz 2.5"
4 I.CC"Podaj numer kanalu, srednice materialu i ilosc elementow czynnych",nk,sm,ie
5 IF nk>=.02 AND nk<=.04 THEN f=nk*700+36.93:G.14
6 IF nk>=.05 AND nk<=.11 THEN f=nk*700+142.98: G.14
7 IF nk=1 THEN f=52.92:G.14
8 IF nk=2 THEN f=62.41:G.14
9 IF nk=2.5 THEN f=70:G.14
10 IF nk>=3 AND nk<=5 THEN f=nk*8+56.44:G.14
11 IF nk>=6 AND nk<=12 THEN f=nk*8+130.47:G.14
12 IF nk>=21 AND nk<=60 THEN f=nk*8+306.48:G.14
13 CLS:P.CC"Nie masz rozeznania w rozdziale kanalow telewizyjnych na poszczegolne panstwa.":G.4
14 ls=300000/f:b=INT(ls/sm):r=

```
INT(.5*ls):u=ie-1:p=.1:gs=
   10.5*LOGie:a=gs/20:wz=
   10^a:rg=5*sm:lr=
   INT(.25*ls)
15 IF b<=40 THEN P.CC"Przy-
   jaleszbyt duza srednice
   materialu."CC"C:G.4
16 IF b>=41 AND b<=100
   THENv=INT((b-41)/10+1):
   k=v*.001+.927:q=v*.
   12+3.38
17 IF b>=101 AND b<=190
   THEN v=INT((b-101)/10+1)
   :k=v*.0012+.934:q=v*.
   095+4.125
18 IF b>=191 AND b<=390
   THEN v=INT((b-191)/10
   +1): k=v*.0004+.9456:q=
   v*.045+5.005
19 IF b>=391 THEN P."'Przyja-
   les zbyt mala srednice ma-
   terialu."CC:G.4
20 w=INT(1500000/f*k):d(1)=
   INT(w-w/20):df=.41*f/q
21 FOR t=1 TO 10:d(t+1)=INT(d
   (t)-d(t)/50):N.t
22 FOR c=1 TO 11:l(c)=INT(p*
   ls):p=p+.02:N.c:CLS
23 IF nk>=.02 AND nk<=.11
   THEN P.CC" Parametry an-
   teny typu Yagi na kanal Nr
   "; nk*100;" w/g CCIR":G.26
24 IF nk=2.5 THEN P.CC" Parame-
   try anteny typu Yagi do od-
   bioru stacji radiowych
   UKF"C:G.26
25 P.CC" Parametry anteny typu
   Yagi na kanal Nr ";nk;" w/g
   OIRT"
26 P.'STRI.39,"%")
27 P."'Ilosc elem. ";ie;" Sred.mat.
   ";sm
28 P.C" Wymiary podane sa w
   milimetrach."
29 C.2:P."'Wibrator po zgieciu
   powinien miec ";w:C.3
30 P."R=";r;" Lr= ";lr
31 FOR n=1 TO u:sd=sd+d(n):sl
   =sl+l(n):P."D";n;" = "; d(n);
   "L";n;" = "; l(n):N.n
32 P." Promien giecia wibratora
   r= ";rg
33 P." Dlugosc calkowita anteny
   l= ";sl+lr+5
34 P." Plaszczyzna naporu
   wiatru = ";(2*w+r+sd)
   *sm/100;"cm.kw."'STRI.39,
   "=")
35 @%=0102020A:VDU23,
   96,10734;11306;26680;
   26792;
36 P." Zysk energet. bedzie
   ";gs;" dB"
37 P." Daje to wzmocnienie syg.
   ";wz;" razy"
38 P." Szer. przenosz. pasma ";df
   ;" MHz" 39@ %=0000090A:
   END
```

Ustalanie kolorów tła, ramki i kursora (Commodore 64)

Szanowna Redakcjo!

Przesyłam krótki, liczący 36 bajtów program w języku maszynowym, napisany dla Commodore 64. Umożliwia on proste i szybkie ustalenia kolorów tła, ramki i kursora. Program zajmuje obszar pamięci \$C000 – \$C023 i nie jest relokowalny. Zastępuje instrukcje POKE 53281,X, POKE 53280,Y, POKE 646,Z. Format, w jakim należy przesyłać dane to:

SYS 49152,X,Y,Z, gdzie X to kolor tła, Y kolor ramki, Z kolor kursora.

Lech Mazur
Nowy Targ

```
10 REM KOLORY
20 FOR T=49152 TO 49187:
   READ A: X=X+A
30 Poke T,A: NEXT
40 IF X=4741 THEN PRINT
   "GOTOWE!": END
50 PRINT "ZLE DANE": END
60 DATA 032,027,192,132,253,
   032,027,192
70 DATA 132,254,032,027,192,
   140,134,002
80 DATA 165,254,141,032,208,
   165,253,141
90 DATA 033,208,096.032,253,
   174,032,158
100 DATA 173,076,170,177
```

Dzielenie z dużą dokładnością (Atari XL/XE)

Jedną z wad języków programowania jest mała dokładność obliczeń. Przeważnie wynosi 6-9 miejsc po przecinku. Nie zawsze jest to wartość wystarczająca.

Chciałbym przedstawić swój program, który wykonuje dzielenie z dokładnością do n-tej liczby po przecinku. Po wpisaniu i uruchomieniu programu należy podać, z jaką dokładnością chcemy wykonywać obliczenia. Następnie wprowadzamy dzielnik i dzielną, muszą to być liczby całkowite większe od zera. Program napisany jest w języku Turbo Basic XL, jednak praktycznie bez zmian (pierwszy wiersz z instrukcją Input "...") może być stosowany w standardowym Basicu. Proponuję jednak, ze względu na szybkość, stosować Turbo Basic lub lepiej jego kompilator.

```
   0 ? :?
10 DIM D1$(100),D2$(100),A$(100)
20 INPUT "PODAJ DOKLAD-
   NOSC ",DOK
30 INPUT "PODAJ DZIELNA ",D1
40 INPUT "PODAJ DZIELNIK ",D2
45 PRINT
50 PRINT D1;"/";D2;"=";
60 IF D1>D2 THEN 500
70 D1$=STR$(D1):D2$=STR$(D2)
80 DL1=LEN(D1$):DL2=LEN
   (D2$)
90 A$=D1$
100 PRINT "0.";
110 IF VAL(A$)<D2 THEN 150
120 A$(LEN(A$)+1)="0"
130 ? "0";
140 GOTO 110
150 REM
160 GOTO 710
500 D1$=STR$(D1):D2$=STR
    (D2):A$=""
510 DL1=LEN(D1$):DL2=LEN(D2$)
520 A$=D1$(1,DL2):DLX=LEN(A$)
530 X1=VAL(A$)
540 IF X1<D2 THEN 800
545 DZ1=DL1-DLX
550 W1=INT(X1/D2)
560 PRINT W1;
570 A$=STR$(X1-W1*D2)
580 DZ1<1 THEN GOTO 700
590 A$(LEN(A$)+1)=D1$(DLX
    +1,DLX+1)
600 DZ1=DZ1-1:DLX=DLX+1
610 X1=VAL(A$)
620 GOTO 550
630 RUN
700 PRINT ".";:D=0
710 A$(LEN(A$)+1)="0"
720 X1=VAL(A$)
730 W1=INT(X1/D2)
740 PRINT W1;
750 A$=STR$(X1-W1*D2)
760 D=D+1:IF D=DOK THEN RUN
770 GOTO 710
800 REM
810 A$(LEN(A$)+1)=D1$(DLX
    +1,DLX+1)
820 DLX=DLX+1:X1=VAL(A$)
830 GOTO 545
```

Może któryś z Czytelników opracuje specjalny pakiet wykonujący wszystkie podstawowe działania matematyczne z podaną dokładnością.

Maciej Stanusch
Tarnowskie Góry

Ustawienie skosu głowicy oraz podsłuch ładowania programów w Datassette (Commodore C16)

Szanowna Redakcjo!

Nawiązując do artykułu "Regulacja głowicy w Datassette" pana J. Żukowskiego ("Bajtek" 11/87) oraz "Magnetofon komputera C64" pana W. Żurka ("Informik" 1/88), stwierdzamy na podstawie kilkuletnich obserwacji, że różnice w fabrycznym ustawieniu głowic w Datassette są duże (często dochodzą do ok. +/- 0.5 obrotu śruby regulacyjnej). Jest to jedną z przyczyn kłopotów z wczytywaniem programów z magnetofonu. Innymi przyczynami są np. zabrudzona głowica lub uszkodzona (załamana) taśma magnetofonowa.

Proponujemy dwa rozwiązania pozwalające na ustawienie (bez oscyloskopu) skosu głowicy z dokładnością 1/8 obrotu, a więc wystarczającą do wczytania programu nawet w dość wybrednym systemie "turbo".

Rozwiązanie pierwsze (sprzętowe) – możliwe do wykonania przez osoby mające niewielką nawet praktykę w posługiwaniu się lutownicą: należy wlutować w komputerze między linię READ gniazda CASSETTE, a linię AUDIO IN gniazda VIDEO rezystor wartości około 10 kiloomów, jak to pokazano na rys. 1 i 2.

Pozwala to na podsłuch programu ładowanego z taśmy. Dzięki temu łatwo możemy znaleźć początek programu, a ustawienie skosu głowicy polega na takiej regulacji głowicy, by najlepiej przenoszone były wysokie tony i sygnał był najsilniejszy. Rozwiązanie ma dwie zalety: mniej obciąża tor odczytu w magnetofonie niż dodatkowy głośnik, pozwala na proste ustawienie głośności potencjometrem w monitorze lub odbiorniku TV.

Rozwiązanie drugie (programowe): poniższy program pozwala na dokładne ustawienie skosu głowicy.

```
1 REM Test glowicy M. Troj-
  nara 1989
10 TRAP 160: POKE 1,200    ;2B
20 FOR A=819 TO 859: READ
   A$: POKE A,DEC(A$):
   NEXT A                  ;01
30 DATA 78,A9,00,A0,CD,CC,
   1D,FF                   ;95
40 DATA D0,FB,A0,C0,AA,E8,
   F0,14                   ;13
50 DATA C4,01,F0,F9,AA,E8,
   F0,0C                   ;4C
60 DATA C4,01,D0,F9,AA,E8,
   F0,04                   ;48
70 DATA C4,01,F0,F9,86,03,58,
   60                      ;8C
100 COLOR 0,1: COLOR 1,1:
    GRAPHIC 1,1: COLOR 1,8,
    6                      ;C7
105 SCNCLR:CHAR,1,1,"M&R
    T.'89 PROGRAM DO REGU-
    LACJI GLOWICY"         ;00
110 CHAR,11,17,"WYJSCIE -
    RUN/STOP": DRAW,117,134
    TO -1,+1               ;7E
120 FOR A=1 TO 318: FOR B=1
    TO 5                   ;EB
130 DO: SYS 819: LOOP UNTIL
    PEEK (3)               ;4A
140 DRAW,A,PEEK(3)         ;B0
150 NEXT B,A: GOTO 105     ;7F
160 COLOR 1,1,1: COLOR 0,2:
    COLOR 4,15,6: GRAPHI-
    CCLR                   ;A5
```

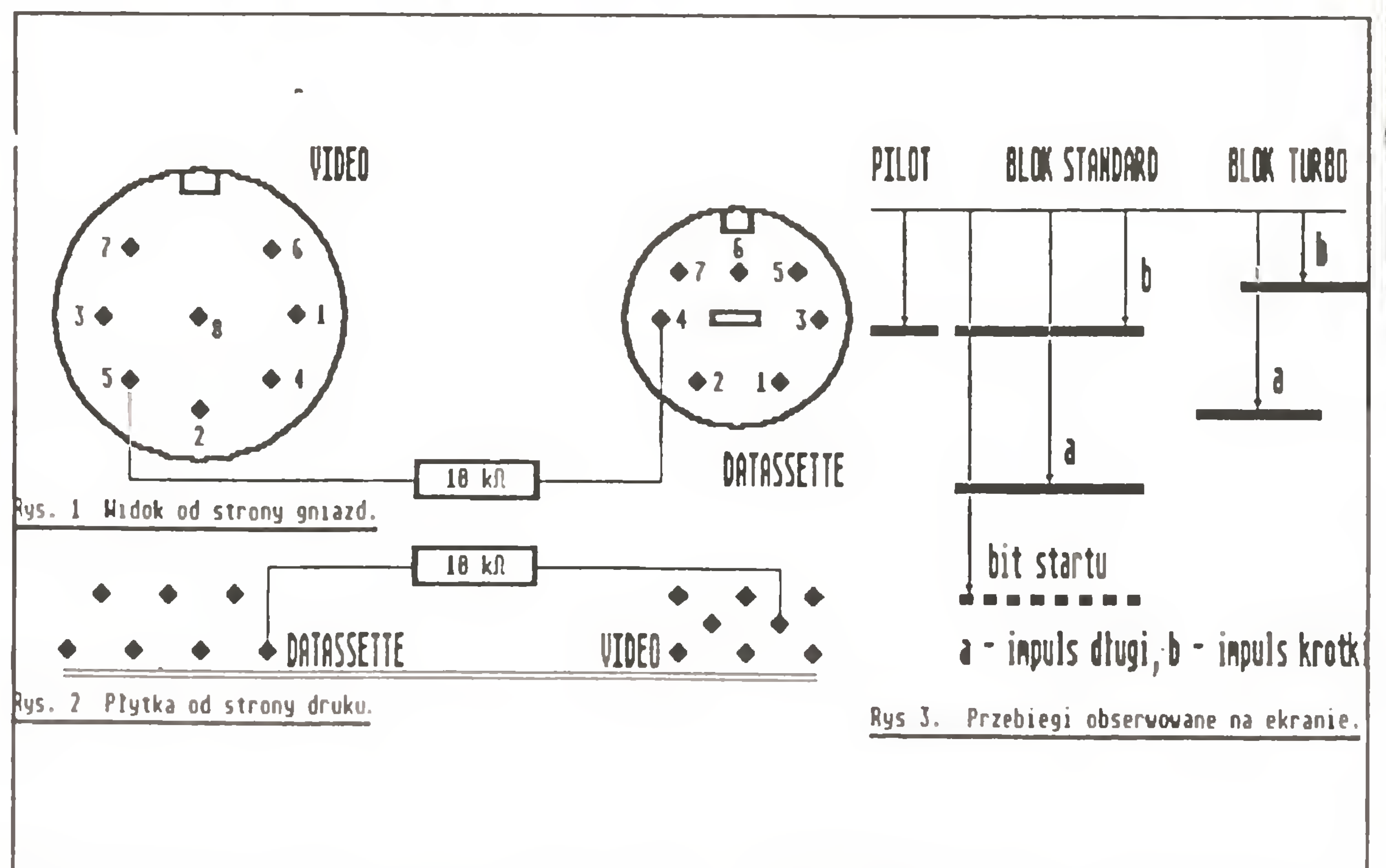
UWAGA: sumy kontrolne uzyskano za pomocą programu WRITEST.

Uruchamiamy powyższy program, wkładamy kasetę, z której chcemy wczytać programy i naciskamy PLAY w Datassette. Przy prawidłowym ustawieniu skosu głowicy przebiegi powinny wyglądać tak, jak pokazano na rys. 3. Gdy linie są "rozmyte", świadczy to o różnym ustawieniu skosu głowicy w naszym Datassette od tego, na którym nagrano program.

Należy zapamiętać pierwotne ustawienie skosu głowicy lub sprawdzić skos według swojej kasety, by nie mieć problemu z wczytaniem programów nagranych przez siebie.

Mimo że nie stwierdziliśmy dotychczas ujemnych skutków takich zabiegów, nie zalecamy przeprowadzania ich bez wyraźnej potrzeby.

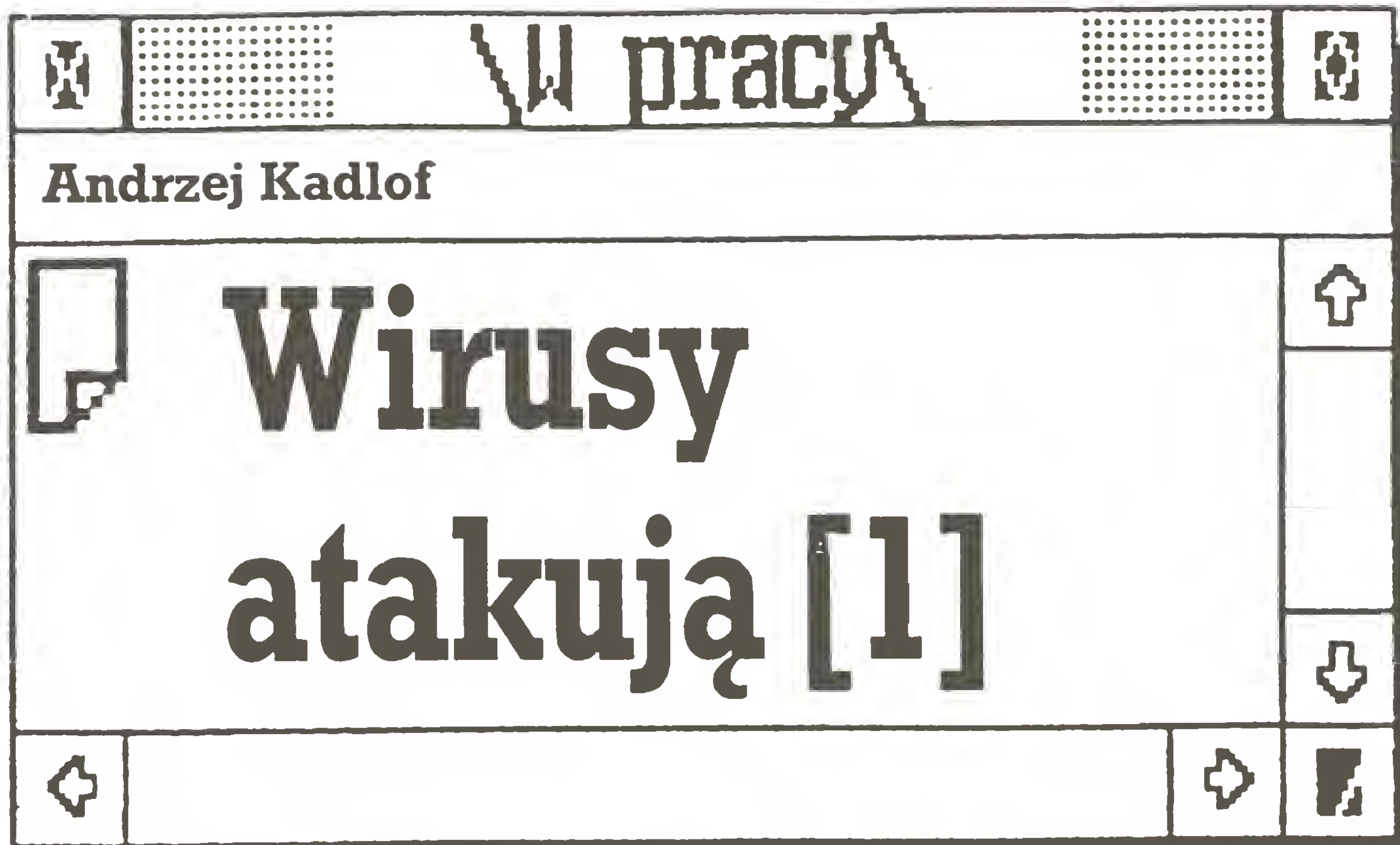
Michał i Ryszard Trojnara
Ostrów Lubelski



Rys. 1 Widok od strony gniazd.

Rys. 2 Płytka od strony druku.

Rys 3. Przebiegi obserwowane na ekranie.



Andrzej Kadlof

Wirusy atakują [1]

Wirusy komputerowe przestały już być w Polsce egzotyczną ciekawostką z dalekiego świata. Pojawiły się naprawdę, siejąc panikę wśród użytkowników, a czasem i zniszczenia w programach. Do momentu napisania tego artykułu udało mi się "schwytać" cztery różne odmiany wirusa. Informacje z rozmaitych niezależnych źródeł sugerują aktywność jeszcze co najmniej dwóch innych, ale dotąd nie zetknąłem się z nimi bezpośrednio.

Analiza kodów wirusów oraz przeprowadzone z nimi eksperymenty wykazują, że środki zaradcze podejmowane przez użytkowników (najczęściej jest to natychmiastowe formatowanie wszystkich zarażonych dysków i dyskietek) są zazwyczaj nieproporcjonalne do faktycznego zagrożenia. Tak brutalne działanie jest co prawda skuteczne, ale wymaga dużo dodatkowej pracy i daje dodatkową satysfakcję wrednym autorom takich programów. Nigdy przy tym nie ma pewności, czy wirus nie uchował się na kopiach zapasowych i czy po odtworzeniu zbiorów na dyskach nie uaktywni się ponownie.

"Polski" wirus i "trzynastka"

Najbardziej znany w Polsce jest tak zwany "polski" wirus (opisany przez Mariusza Deca w 11. numerze "Komputera" z ubiegłego roku). Swoją nazwę zawdzięcza chyba tylko temu, że został jako pierwszy zidentyfikowany i opisany w naszym kraju. Listy do redakcji wskazują, że jest już bardzo szeroko rozpowszechniony i wielu programistów zmuszonych jest do marnowania swojego czasu na pisanie odpowiednich "szczepionek".

Wirus ten najprawdopodobniej pochodzi z Austrii i poza Polską na pewno jest znany w kilku krajach Europy Zachodniej oraz w Czechosłowacji. Jego działanie jest bardzo zbliżone do innego, znanego pod nazwą "trzynastka". Podobieństwa są jednak tylko zewnętrzne i nie można ich traktować jako mutacji.

Wirusy te atakują tylko pliki z rozszerzeniem COM. Doklejają się do nich powiększając ich długość odpowiednio o 658 i 530 bajtów. Bieżący stan ustawienia atrybutów pliku (Hiden, Read Only itd.) nie stanowi dla nich żadnej przeszkody. Oba oszczędzają pliki o długości ponad 64000 bajtów. "Polski" wirus ignoruje również pliki o długości mniejszej od 10, podczas gdy "trzynastka" jest nieco przezor-

niejsza i omija pliki o długości poniżej 256 bajtów. (Zmiana długości krótkich plików szybko rzuca się w oczy!).

Po uruchomieniu, oba wirusy podejmują próbę zarażenia tylko jednego pliku. Jeśli nie znajdą "ofiary", nie robią nic. W razie powodzenia oznakowują w katalogu zainfekowany program w sposób, jaki nie może powstać przypadkowo. "Polski" wirus zmienia czas ostatniej modyfikacji pliku. W miejsce sekund wstawia liczbę 31, co jest interpretowane przez DOS jako 62 sekundy! Z kolei "trzynastka" zmienia datę wstawiając w miejsce numeru miesiąca liczbę 13, Dzięki temu wirusy unikają wielokrotnego atakowania tego samego pliku.

DOS ani żadna z popularnych nakładek nie pokazują czasu ostatniej modyfikacji pliku z dokładnością do sekundy, dlatego system znakowania stosowany przez naszego wirusa pozostaje praktycznie niewidoczny dla użytkownika. Z kolei pojawienie się w katalogu miesiąca o numerze 13 od razu rzuca się w oczy. Wydaje się, że autorom obu wirusów przyświecały różne cele. Świadczy o tym również tak zwana druga funkcja wirusa (pierwszą jest zawsze powielanie własnego kodu, druga to zamierzona złośliwość autora).

"Trzynastka" nie wyrządza krzywdy poza tym, że zwiększa długość plików i zmienia datę ostatniej modyfikacji. Jedynym jej celem jest powielanie własnego kodu. Najwyraźniej autorowi wystarczyła radosna wizja panicznych działań użytkownika, który zorientuje się, że w jego systemie pojawił się wirus. Druga funkcja została ograniczona do sfery psychologicznej i niestety, jak pokazuje życie, jest bardzo efektywna. Autor "polskiego" wirusa nie ograniczył się tylko do straszenia.

Oba wirusy w zarażonym programie zmieniają trzy pierwsze bajty na skok do własnego kodu umieszczanego na samym końcu pliku. Oryginalne bajty są przez nie przechowywane i umożliwiają im, po próbie zakażenia innego pliku, na ich odtworzenie i wykonanie właściwego programu. Użytkownik zazwyczaj nie zauważa, że zainfekowany program wykonał cokolwiek więcej niż robił do tej pory. Niestety średnio co osiem ataków (wykorzystano tu odczyt zegara systemowego) "polski" wirus zmienia pięć pierwszych bajtów (zapamiętując tylko trzy!) na skok do procedury w ROM-ie, powodującej restart systemu operacyjnego. Tak zmodyfikowane programy są już w zasadzie zniszczone i muszą być wymienione na zdrowe kopie.

"Trzynastka" i "polski" wirus istotnie różnią się mechanizmem wyszukiwania kandydatów do zarażenia. Początkowo oba przeszukują katalog roboczy i infekują zdrowe dotąd programy. Po zarażeniu wszystkich "trzynastka" przechodzi do przeszukania katalogu głównego dysku roboczego. Inne katalogi są omijane. "Polski" wirus jest nieco bardziej wyrafinowany. Gdy już zakazi wszystkie programy w katalogu roboczym, przechodzi do przeszukiwania katalogów, które sami podaliśmy systemowi operacyjnemu za pomocą zlecenia PATH. Ta cecha pozwala na łatwe zablokowanie możliwości przemieszczania się go do innych katalogów. Jak zauważył Mariusz Dec, wystarczy w zleceniu PATH podać jako pierwszy jakiś nieistniejący katalog. Wirus ten bowiem kończy przeszukiwanie ścieżek po pierwszym sygnale niepowodzenia.

Jak widać, wirusy, choć nieco kłopotliwe nie stanowią większego zagrożenia. Jeśli mieliśmy trochę szczęścia i "polski" wirus nie zniszczył żadnego pliku typu COM, a jedynie je zmodyfikował, to do czasu dogrania nowych programów tego typu system nasz będzie funkcjonował prawidłowo i nie wykaże żadnych objawów infekcji.

"Spadające znaki ASCII"

Trzeci wirus jest zupełnie odmienny od pozostałych. Jego autor popisał się dużym doświadczeniem i wiedzą daleko wykraczającą poza to, co można znaleźć w łatwo dostępnych opisach systemu operacyjnego PC DOS. Wykazuje on kilka interesujących cech.

Po pierwsze – instaluje się rezydentnie w systemie i pozostaje aktywny aż do czasu restartu systemu operacyjnego komputera.

Po drugie, jest tak zaprogramowany, żeby w sposób widoczny objawiał się jedynie w ostatnim kwartale 1988 roku i podobno tylko na komputerach wyposażonych w kartę EGA (testy przeprowadzone na Herculesie nie wykazały jego aktywności, niepełna analiza kodu sugeruje, że może być również aktywny na karcie CGA). Jego działanie polegało na efektownym "zrzucaniu" z ekranu znaków ASCII.

Po trzecie – rozmnażać się powinien jedynie w systemach bez bateryjnego zegara (czyli takich, w których operator musi sam podawać czas i datę po uruchomieniu systemu operacyjnego). Jeśli leniwy użytkownik popędza instalację systemu PC DOS klawiszem ENTER (przez co automatycznie wprowadza datę 1988-01-01), a "komputer jest zainfekowany", wirus będzie się powielał dalej. Do takich wniosków doprowadziła mnie analiza jego kodu. Niestety w praktyce okazało się, że wirus ten rozmnaża się niezależnie od daty.

W odróżnieniu od poprzednich nie szuka sobie ofiary, a czeka aż użytkownik sam mu ją podsunie. Dokleja się bowiem do kolejno uruchamianych programów typu COM i zwiększa ich długość o 1701 bajtów. Nie stosuje żadnego oznakowywania zainfekowanych plików w katalogach tak, że jedynie zmiany długości plików mogą być ostrzeżeniem dla użytkownika.

Najbardziej jednak zaskakującą jego cechą jest to, że oszczędza komputery znanej również u nas firmy IBM. Przed zainstalowaniem się sprawdza w ROM-ie komputera czy w polach przeznaczonych na identyfikator producenta znajduje się napis: "COPR. IBM". Takie komputery są dla niego nietykalne! Czy taki był komputer autora czy też ...?

Złośliwość tego wirusa jest pośrednia między "polskim" wirusem i "trzynastką". Nie niszczy żadnych zbiorów, ale utrudnia (a właściwie utrudniał w ostatnim kwartale 1988 roku) pracę z komputerem. Niestety nie można tego samego powiedzieć o następnym paskudztwie.

"Dowcipas"

Czwarty program (nie spotkałem się jeszcze z żadną jego potoczną nazwą) łączy z wirusami to, że powiela się w systemie. W odróżnieniu jednak od klasycznych wirusów nie usiłuje się ukrywać i od razu demonstruje swoją działalność. Po wywołaniu zainfekowany program zakaża inny i on zamiast normalnie działać wyświetla na ekranie jakiś komunikat. Tych ma cały zestaw: od chamskich, poprzez udające oryginalne komunikaty systemu, do nawet dowcipnych typu: "Ostrzeżenie! W stacji A: wykryto dwie dyskietki!", "UWAGA! Wykryto wodę w koprocesorze" lub "Nie męcz mnie!". Rozbawiony użytkownik może zechcieć ponownie uruchomić program i losowo pojawiające się komunikaty dadzą mu przez moment zabawę. Przestanie to być zabawne, gdy się okaże, że pozostałe pliki typu EXE potrafią wyświetlać już tylko takie komunikaty.

Zainfekowany program rozpoczyna działanie od szukania innego kandydata do zarażenia. Przeszukuje w tym celu dysk C: (zazwyczaj jest to twardy dysk), następnie dysk A: i w końcu bieżący. Szuka na nich plików z rozszerzeniem EXE i DBF. Infekcja pliku oznacza niestety jego bezpowrotne zniszczenie, bo wirus kopiuje się w całości na jego początku (ponad 10 Kb).

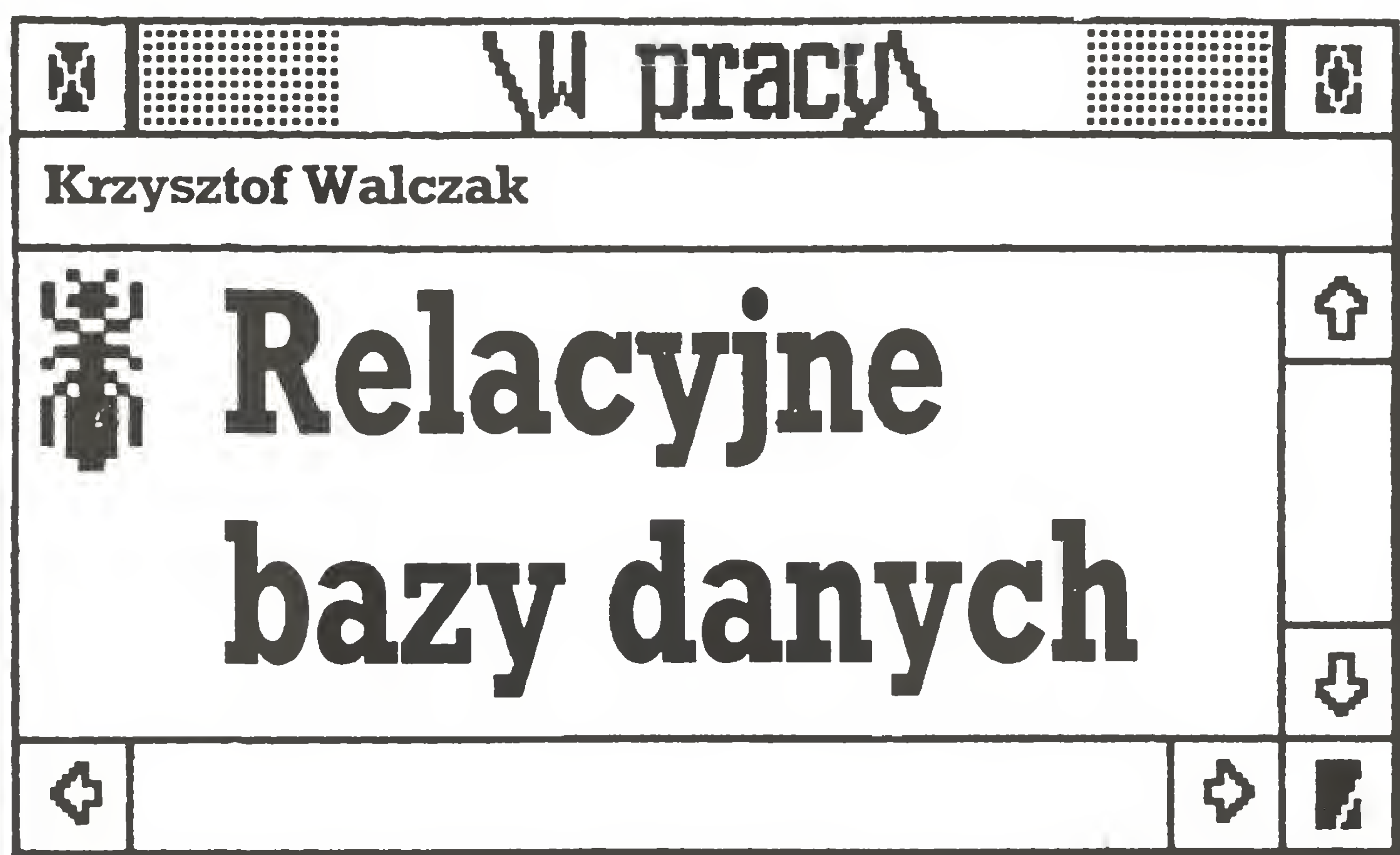
Z uwagi na swoją agresywność ma on niewielkie szanse na rozprzestrzenianie się poza ogólnodostępnymi sieciami modemowymi. Z tych względów jego żywot w Polsce będzie raczej krótki i jeśli pozostanie na jakichś dyskach, to raczej jako ciekawostka lub jako dowód skrajnej niefrasobliwości (lub ignorancji) użytkownika.

Jest on jednocześnie przykładem, że do robienia bezinteresownych świństw innym użytkownikom niepotrzebna jest żadna wyrafinowana wiedza. Ten program napisany został najprawdopodobniej w Turbo Pascalu 4.0 lub 5.0. Cała informacja potrzebna do jego stworzenia znajduje się w firmowej instrukcji obsługi (może właśnie dlatego ten wirus nie potrafi maskować swego wtargnięcia do systemu) i należy do podstawowej wiedzy nieco bardziej zaawansowanego programisty.

Analiza kodów cudzych programów, a zwłaszcza pisanych w językach wyższego pozimu, jest zajęciem bardzo żmudnym, a w przypadkach kodów wirusów wręcz ryzykownym (w czasie pierwszych eksperymentów "polski" wirus, na skutek mojego zmęczenia lub niedostatecznej ostrożności, zdołał mi się kilkakrotnie wymknąć spod kontroli programu śledzącego i zainstalować na twardym dysku!). Stąd też nie wszystkie szczegóły działania dwóch ostatnich wirusów są dla mnie do końca jasne. Sądzę jednak, że powyższe dane umożliwią każdemu je zidentyfikować, jeśli pojawią się w systemie.

Obawiam się, że powyższa lista grasujących po naszych komputerach wirusów nie jest kompletna i co gorsza będzie się szybko powiększać. Jest to cena jaką będziemy płacić za pirackie kopiowanie programów z wszelkich dostępnych źródeł.

Nie chcę tu opisywać dalszych przypadków znanych mi jedynie z cudzych opowiadań (plotek?). Nie wydaje mi się, by trzeba było dodatkowo kogokolwiek straszyć. Jeśli jednak komuś z Czytelników uda się "schwytać" jakiegoś wirusa, innego od tu opisanych, to bardzo proszę o przesłanie do redakcji kopii przynajmniej jednego zainfekowanego pliku celem jego analizy i ostrzeżenia innych użytkowników.



Baza danych jest to uporządkowany zbiór danych przechowywany w pamięci pomocniczej komputera. System bazy danych można zdefiniować jako bazę danych wraz z oprogramowaniem umożliwiającym operowanie danymi zawartymi w bazie.

Obecnie bazy danych są wykorzystywane prawie wszędzie: w bankach, administracji, na uczelniach, w dużych i ostatnio coraz mniejszych przedsiębiorstwach. Należy tylko zdawać sobie sprawę z tego, że mikrokomputerowe bazy danych nie wszędzie mogą być wykorzystywane. W wielu zastosowaniach mikrokomputer jest niewystarczający.

Na świecie jest obecnie bardzo dużo systemów narzędziowych umożliwiających tworzenie oprogramowania bazy danych (większość dla mikrokomputerów IBM PC). W Polsce największą popularność zdobył sobie system dBase firmy Ashton-Tate z USA umożliwiający założenie i wykorzystywanie relacyjnej bazy danych. Początkowo był to dBase II, następnie dBase III i dBase III plus, a ostatnio dBase IV (jeszcze w Polsce mało znany). System dBase umożliwia pracę z bazą danych w trybie interakcyjnym oraz konstrukcję programów, które wykonują operacje na bazie danych i są interpretowane – tzn. każda instrukcja programu jest tłumaczona na kod wynikowy i od razu wykonywana.

Równolegle z systemem dBase rozwijany był kompilator Clipper firmy Nantucket umożliwiający kompilowanie programów napisanych w języku dBase (system dBase IV ma swój własny kompilator) oraz dostarczający pewnych środków programowych zwiększających efektywność pracy. Ze względu na znaczne rozszerzenia dostępne w kompilatorze Clipper w stosunku do języka dBase można mówić o języku Clipper jako niezależnym języku umożliwiającym programowanie systemów relacyjnych baz danych. Programy napisane w języku Clipper są kompilowane za pomocą kompilatora o tej samej nazwie, następnie konsolidowane (w trakcie konsolidacji dołączane są niezbędne funkcje języka), a wykonywany jest otrzymany program wynikowy. Należy podkreślić, że skompilowanie programu umożliwia kilkakrotne przyspieszenie pracy programu, co pozwala na wykorzystanie takiego programu przez osoby nie mające prawie żadnego przeszkolenia informatycznego oraz uniemożliwia przypadkowe zniszczenie zawartości bazy danych. Często też rzeczą niebagatelną jest możliwość sprzedaży użytkownikowi wyłącznie programu wynikowego, bez programów źrodłowych.

Relacyjne bazy danych

Przed zdefiniowaniem relacyjnej bazy danych wprowadzone zostanie pojęcie relacji znane z teorii mnogości. Relacja jest podzbiorem iloczynu kartezjańskiego dziedzin $D_1, D_2, ..., Dn$. Dziedzina jest zbiorem pewnych wartości, np. może to być zbiór liczb całkowitych, zbiór liczb z przedziału <0,5>, zbiór znaków o długości nie większej niż 10 itp. Iloczyn kartezjański oznacza się następująco $D_1 * D_2 * ... * D_n$.

Zawiera on n-tki $(d_1, ..., d_n)$ nazywane krotkami takie, że $d \in D_1$, $d_2 \in D_2, ..., d_n \in D_n$.

Przykład 1.

Niech $D_1 = \{a,b,c\}$, $D_2 = \{x,y\}$

$D_1 * D_2 = \{(a,x),(a,y),(b,x),(b,y),(c,x),(c,y)\}$

Poniżej podano dwa przykłady relacji będące podzbiorami iloczynu kartezjańskiego $D_1 * D_2$:

$R1 = \{(a,x),(b,x),(c,x)\}$

$R2 = \{(a,x),(a,y),(b,y)\}$

Analogicznie jak dla iloczynu kartezjańskiego elementy relacji będą nazywane krotkami.

Relacyjna baza danych jest to baza danych, w której dane przedstawiane są w formie relacji. Relację można przekazać w postaci tablicy (tablica jest strukturą danych powszechnie wykorzystywaną w programowaniu). Każdy wiersz tablicy można interpretować jako krotkę relacji. Kolumnom tablicy można nadawać nazwy, które są atrybutami relacji. Zbiór atrybutów nazywamy schematem relacji.

W terminologii systemów mikrokomputerowych relacji odpowiada plik, krotce – rekord, a atrybutowi – pole.

Przykład 2.

Rozważmy relację, której atrybutami są nazwisko, imię, wiek. Relację tę można zapisać następująco:

PRAC<nazwisko, imię, wiek>

gdzie PRAC jest nazwą danej relacji.

Poniżej przedstawiono trzy krotki relacji PRAC:

<Kowalski, Jan , 36>

<Tomaszewski, Wojciech, 40>

<Wiśniewski, Marek, 50>

Relację tę można przedstawić w postaci tablicy w sposób następujący:

nazwisko	imię	wiek
Kowalski	Jan	36
Tomaszewski	Wojciech	40
Wiśniewski	Marek	50

Zauważmy, że jeżeli poszczególne kolumny mają przypisane nazwy (atrybuty), to kolejność zapisywania kolumn jest nieistotna.

Przykład 3.

Poniżej podano dwa różne przedstawienia tej samej relacji w postaci dwóch tablic.

Tablica I.

mistrz	wydział	materiał	ilość	cena
Kowalski	1	deski	3	1000
Tomaszewski	2	śruby	10	10

Tablica II.

wydział	mistrz	materiał	cena	ilość
1	Kowalski	deski	1000	3
2	Tomaszewski	śruby	10	10

Głównym problemem przy projektowaniu relacyjnych baz danych jest właściwy wybór schematu relacji. Okazuje się, że jedne schematy zapewniają bardziej korzystne właściwości bazy danych niż inne. Problem ten ilustruje następujący przykład.

Przykład 4.

Rozważmy bazę danych zawierającą informacje o czytelnikach i wypożyczonych przez nich książkach w bibliotece o następującym schemacie:

BIBL<nr, nazwisko, adres, ksiażka, data>

gdzie:

- atrybut nr – określa numer czytelnika,
- atrybut adres – określa adres czytelnika,
- atrybut książka – określa nazwę wypożyczonej książki,
- atrybut data – określa datę wypożyczenia książki.

Poniżej przedstawiono 5 krotek rozważanej relacji zapisanej w tablicy:

nr	nazwisko	adres	ksiażka	data
45	Kowalski	Goldoniego 4 m.5	Pascal	1988.12.14
67	Mostowski	Szekspira 2 m. 3	Clipper	1987.09.12
78	Lipski	Dantego 3 m.7	Informatyka	1979.12.14
78	Lipski	Dantego 3 m.7	Język C	1979.10.12
45	Kowalski	Goldoniego 4 m.5	Ada	1981.10.11

Przy wykorzystywaniu bazy danych o przedstawionym wyżej schemacie relacji pojawia się kilka problemów.

1. Redundancja. Baza danych zaprojektowana w powyższy sposób zawiera redundancję, to znaczy, że pewne informacje się powtarzają. Jeżeli czytelnik wypożyczył kilka książek, to w odpowiednich dotyczących go krotkach powtarza się jego numer, nazwisko i adres.

2. Anomalie przy aktualizacji (niespójność bazy danych). W przypadku zmiany adresu pana Kowalskiego i uaktualnienia tego adresu wyłącznie w pierwszej krotce pojawia się niespójność bazy danych. Adres Kowalskiego w różnych krotkach relacji jest różny. Aktualizacja adresu Kowalskiego we wszystkich krotkach bazy danych zawierającej informację o nim jest możliwa, ale zbyt pracochłonna.

3. Anomalie przy usuwaniu. W przypadku likwidacji informacji o wszystkich książkach wypożyczonych przed rokiem 1980 można stracić informację o jakimś czytelniku.

4. Anomalie przy wstawianiu. W bazie danych zaprojektowanej w powyższy sposób nie można zapisać adresu czytelnika, który nie wypożyczył ani jednej książki.

W przedstawionej wyżej bazie danych wymienionych problemów można by uniknąć zastępując schemat relacji dwoma schematami o postaci:

CZYTELNIK<nr, nazwisko,adres> oraz

KSIĄŻKI<nr, książka, data>

Baza danych przedstawiona w powyższym przykładzie zawierałaby wtedy dwa zbiory (dwie tablice):

Tablica I.

nr	nazwisko	adres
45	Kowalski	Goldoniego 4 m. 5
67	Mostowski	Szekspira 2 m. 3
78	Lipski	Dantego 3 m. 7

Tablica II.

nr	książka	data
45	Pascal	1988.12.14
67	Clipper	1987.09.12
78	Informatyka	1979.12.14
78	Język C	1979.10.12
45	Ada	1970.10.12

Baza ta nie zawiera redundancji, niespójności i anomalii.

Programowanie systemów baz danych w języku Clipper

Kompilator Clipper można wykorzystać w dwojaki sposób. Pierwszy polega na programowaniu wyłącznie w języku Clipper. Pliki baz danych tworzone są za pomocą programu dostarczanego wraz z kompilatorem. Drugi sposób polega na utworzeniu plików baz danych w systemie dBase, napisaniu i uruchomieniu programów w sposób interakcyjny i dopiero wtedy skompilowaniu ich za pomocą kompilatora Clipper. Należy jednak podkreślić, że nie można wtedy wykorzystać różnych funkcji dostępnych w języku Clipper, a niedostępnych w systemie dBase.

W niniejszym artykule zasygnalizowany zostanie wyłącznie pierwszy sposób wykorzystywania kompilatora Clipper. Programowanie systemów użytkowych baz danych, tzn. systemów pozwalających w automatyczny sposób aktualizować bazę danych, dodawać rekordy, usuwać wybrane rekordy oraz otrzymywać z bazy danych żądane informacje różni się od programowania innych zagadnień, niemniej podstawowe zasady i reguły pozostają bez zmiany. Pierwszym krokiem jest przeprowadzenie dokładnej analizy problemu, który mamy zaprogramować. Analiza ta powinna doprowadzić do sformułowania metody rozwiązania problemu – tzn. algorytmu. Algorytm jest to metoda rozwiązania pewnego problemu sformułowana w skończonej liczbie kroków elementarnych. Definicja kroku elementarnego zależy od tego na jakim stopniu szczegółowości rozpatruje się dany algorytm. W jednym algorytmie krokiem elementarnym może być np. żądanie usunięcia danego rekordu, a w drugim może to być np. usunięcie pewnego zbioru rekordów w danym pliku bazy danych.

W celu zaprojektowania efektywnego i spełniającego oczekiwania użytkownika systemu bazy danych należy rozważyć kilka aspektów. Po pierwsze, trzeba odpowiednio zaprojektować samą bazę danych.

Po zaprojektowaniu bazy danych należy zastanowić się nad tym, jakie funkcje powinien spełniać system użytkowy. Na ogół system taki powinien umożliwiać w sposób automatyczny aktualizację, usuwanie, dodawanie wybranych rekordów w dowolnym pliku bazy lub jednocześnie w kilku plikach powiązanych zadaną zależnością. System powinien też umożliwiać wyszukanie żądanej informacji o danych przechowywanych w bazie danych. W tym celu należy uzgodnić z użytkownikiem systemu, jakie informacje będą mu potrzebne i w ten sposób zaprojektować menu systemu, aby wszystkie żądane informacje można było uzyskać po wybraniu odpowiedniej opcji menu. Ponadto system powinien drukować zestawienia zbiorcze o danych przechowywanych w bazie.

Menu systemu może być zaprojektowane w różny sposób. Może to być menu zagnieżdżone (takie zostało wykorzystane w przykładzie zamieszczonym poniżej), może to być menu wieloetapowe, a może też być menu globalne. W menu tym wszystkie możliwe opcje są dostępne z głównego ekranu. Na ekranie jest umieszczony w wydzielonym oknie fragment przetwarzanego pliku bazy danych, tzn. tylko kilka rekordów i trzy wybrane pola pliku. W drugim oknie ekranu jest umieszczona zawartość bieżącego rekordu a w trzecim wszystkie możliwe opcje menu.

Kolejnym krokiem jest zaprojektowanie postaci ekranów tzn. rodzaju i postaci informacji wyświetlanych jednorazowo na ekranie. Należy to zrobić szczególnie starannie, pamiętając o kasowaniu menu wyświetlonych poprzednio na ekranie, ponieważ ekran jest podstawowym narzędziem do komunikacji z użytkownikiem. Ostatnią czynnością jest zaprojektowanie rodzaju informacji pomocniczej o systemie, ukazującej się na ekranie po naciśnięciu klawisza F1.

Programowanie menu systemu w języku Clipper

W języku Clipper dostępne są instrukcje i funkcje pozwalające w efektywny sposób zaprogramować menu systemu. Są to następujące instrukcje **MENU TO, @... PROMPT, SET MESSAGE TO** oraz funkcja **ACHOICE**. Sposób wykorzystania instrukcji MENU TO oraz instrukcji @... PROMPT ilustruje poniższy przykład.

Przykład 5.

```
*********************************************************
*     Program ilustrujacy wykorzystanie instrukcji      *
*     pozwalających utworzyć proste menu systemu        *
*********************************************************
CLEAR
kontr1 = 1          && zmienna kontrolna menu
SET MESSAGE TO 24
DO WHILE kontr1 != 0
  @7,0 PROMPT "WYBOR_GLOWNY_1" MESSAGE ;
        "Informacja pierwsza (nacisnij enter)"
  @7,20 PROMPT "WYBOR_GLOWNY_2" MESSAGE ;
        "Informacja druga (nacisnij enter)"
  @7,40 PROMPT "WYBOR_GLOWNY_3" MESSAGE;
        "Informacja trzecia (nacisnij enter)"
  MENU TO kontr1
  IF kontr1 !=   0 && klawisz Esc – wyjście z menu
    DO CASE
      CASE kontr1 = 1
        DO Wydruk
      CASE kontr1 = 2
        DO Wydruk
      CASE kontr1 = 3
        DO Wydruk
    ENDCASE
  ENDIF
ENDDO
```

Procedura Wydruk zamieszczona jest w następnym przykładzie.

Instrukcja SET MESSAGE TO podaje, w którym wierszu wyświetlona będzie informacja zawarta w instrukcji PROMPT w opcji MESSAGE.

Instrukcje PROMPT wyświetlają na ekranie odpowiednie opcje menu. W przykładzie wszystkie opcje wyświetlone są w wierszu 7. odpowiednio od zerowej, dwudziestej i czterdziestej kolumny. Opcje menu można wyświetlić w dowolnym miejscu na ekranie (również w różnych wierszach)

Instrukcja MENU TO powoduje uaktualnienie danego menu. Po wybraniu odpowiedniej opcji zmienna kontrolna kontr1 otrzymuje odpowiednią wartość, co pozwala następnie na wywołanie odpowiedniego programu lub procedury w zależności od wartości zmiennej kontr1.

Istnieje również możliwość projektowania "menu" zagnieżdżonego, tzn. takiego, że po wyborze jednej z możliwości menu głównego wyświetla się lista możliwości menu wewnętrznego. W tym celu wykorzystuje się instrukcje **SET KEY TO** oraz **KEYBOARD**. Instrukcja **SET KEY TO** pozwala na wywołanie odpowiedniej procedury w zależności od ostatnio naciśniętego klawisza. Natomiast instrukcja **KEYBOARD** zapełnia bufor klawiatury żądanymi znakami. Instrukcje te, współpracując z sobą, umożliwiają po naciśnięciu jednego klawisza zapełnienie bufora klawiatury kilkoma znakami, co powoduje wykonanie kilku operacji.

> 32

31 <

Przykład 6.

```
*************************************************************
*     Program ilustrujący wykorzystanie instrukcji          *
*     pozwalających utworzyć dwupoziomowe menu systemu*
*************************************************************
CLEAR
kontr1 = 1      && zmienna kontrolna menu głownego
kontr2 = 1      && zmienna kontrolna menu wewnętrznego
pom = 0         && zmienna kontrolujaca wywołanie procedur
                && obsługi kursora
SET MESSAGE TO 24
DO WHILE kontr1 != 0
  @ 7,0 PROMPT "WYBÓR_GŁÓWNY_1" MESSAGE ;
  "Informacja pierwsza (naciśnij enter)"
  @ 7,20 PROMPT "WYBÓR_GŁÓWNY_2" MESSAGE ;
  "Informacja druga (naciśnij enter)"
  @ 7,40 PROMPT "WYBÓR_GŁÓWNY_3" MESSAGE;
  "Informacja trzecia (naciśnij enter)"
  MENU TO kontr1
IF kontr1 != 0                && klawisz esc – wyjście z menu
  DO CASE
    CASE kontr1 = 1
      DO Wybor1 WITH kontr1
    CASE kontr1 = 2
      DO Wybor2 WITH kontr1
  CASE kontr1 = 3
      DO Wybor1 WITH kontr1
  ENDCASE
  ENDIF
ENDDO
● Procedura obsługująca pierwszy wybór z menu głównego
PROCEDURE Wybor1
PARAMETER mn
        @ 9, (mn-1)*20 PROMPT 'Wybor_1_1'
        @ 10, (mn-1)*20 PROMPT 'Wybor_1_2'
● Ustawienie procedur obsługi kursora
SET KEY 19 TO Lewa
SET KEY 4 TO Prawa
MENU TO kontr2
● Odwołanie procedur obsługi kursora
SET KEY 19 TO
SET KEY 4 TO
IF kontr2 <> 0 .AND. pom = 0
  DO CASE
    CASE kontr2 = 1
      * Uwaga: zamiast procedury wydruk można wywołać
      * dowolny program lub procedurę obsługującą dany
      * wybór z menu
        DO Wydruk
  CASE kontr2 = 2
        DO Wydruk
  ENDCASE
ENDIF
pom = 0
@ 9,0 CLEAR
RETURN
● Procedura obsługująca drugi wybór z menu głównego
PROCEDURE Wybor2
PARAMETER mn
@ 9, (mn-1)*20 PROMPT 'Wybor_2_1'
@10,(mn-1)*20 PROMPT 'Wybor_2_2'
@11,(mn-1)*20 PROMPT 'Wybor_2_3'
● Ustawienie procedur obsługi kursora
SET KEY 19 TO Lewa
SET KEY 4 TO Prawa
MENU TO kontr2
● Odwołanie procedur obsługi kursora
SET KEY 19 TO
SET KEY 4 TO
if kontr2 <> 0 .AND. pom = 0
DO CASE
  CASE kontr2 = 1
    DO Wydruk
  CASE kontr2 = 2
    DO Wydruk
  CASE kontr2 = 3
    DO Wydruk
ENDCASE
endif
pom = 0
@ 9,0 CLEAR
RETURN
● Procedura wydruku informacyjnego
PROCEDURE Wydruk
SAVE SCREEN
CLEAR
@  0,0 TO 24,79 DOUBLE
@ 10,10 SAY 'Uwaga: zamiast procedury wyświetlajacej ten'
@ 11,10 SAY 'wydruk można wywołać dowolną procedurę
@ 12,10 SAY 'obsługującą dany wybór z menu'
@ 23,3 SAY 'NACIŚNIJ ENTER'
WAIT ''
RESTORE SCREEN
RETURN

● procedura obsługująca lewą strzałkę
PROCEDURE Lewa
* zapamiętywany jest znak powrotu strzałki w górę i powrotu
KEYBOARD chr(13) + chr(5) + chr(13)
pom = 1            && zapamiętanie faktu wywołania procedury
RETURN

● procedura obsługująca prawą strzałkę
PROCEDURE Prawa
* zapamiętywany jest znak powrotu strzałki w dół i powrotu
KEYBOARD chr(13) + chr(24) + chr(13)
pom = 1            && zapamiętanie faktu wywołania procedury
RETURN
```

Program główny nie różni się prawie od programu zamieszczonego w poprzednim przykładzie, z tym że występują tu jeszcze dwie zmienne: **kontr2** zmienna kontrolna menu wewnętrznego oraz zmienna **pom** kontrolująca wywołanie procedur Lewa i Prawa.

W zależności od dokonanego wyboru w menu głównym (w programie głównym) można wywołać dwie procedury **Wybor1** i **Wybor2**. W każdej z tych procedur istnieje wewnętrzne menu ze zmienną kontrolną kontr2. Przed wykonaniem instrukcji MENU TO wykonywane są instrukcje **SET KEY 19 TO Lewa** oraz **SET KEY 4 TO Prawa**. Instrukcje te powodują, że w momencie naciśnięcia klawisza lewej strzałki (kod ASCII 19) lub klawisza prawej strzałki (kod ASCII 4) wywoływane są odpowiednio procedury Lewa i Prawa. Procedury te zapełniają bufor klawiatury odpowiednio znakami powrotu, strzałki w górę, powrotu lub znakami powrotu, strzałki w dół, powrotu. Powoduje to, że instrukcja MENU TO wykonywana jest tak, jakby użytkownik nacisnął kolejno trzy klawisze. Znak powrotu powoduje powrót do programu głównego, znak strzałki w dół lub w górę powoduje przejście w lewo lub w prawo w menu głównym, a znak powrotu powoduje wywołanie odpowiedniej procedury rozwijającej menu wewnętrzne.

Zmienna pomocnicza pom wykorzystywana jest do zapamiętania faktu wywołania procedur Lewa lub Prawa po to, aby z procedur Wybor1 lub Wybor2 przejść od razu do programu głównego.

W programie głównym wywoływane są dwie procedury Wybor1 i Wybor2, mimo że menu zawiera trzy opcje. Oczywiście liczba wywoływanych procedur może być dowolna.

W niektórych systemach użytkowych może być pożyteczne napisanie procedury, która w sposób automatyczny rozwijałaby opcje menu. Jednocześnie można sprawdzić czy użytkownik nie nacisnął jakiegoś klawisza sterującego pewnymi funkcjami np. aktualizacją rekordów lub usuwaniem rekordów.

Przykład 7.

```
* procedura wyświetlająca proste menu systemu
PROCEDURE MENU
PARAMETERS x,y,w,h,opcje, kontr,zn, wyn
* x,y – współrzędne lewego górnego rogu
* w – szerokość jednej opcji
* h – liczba opcji
* opcje – tekst zawierający kolejno wszystkie opcje
* zn – kod wyróżnionego znaku
* wyn – zmienna logiczna informująca o tym czy naciśnięto
*        dany znak
PRIVATE x1,y1,x2,y2,s
wyn = .F.
x1 = x + 1
y1 = y + 1
x2 = x + h + 1
y2 = y + w + 1
s = 1
* rozpisanie kolejno wszystkich opcji umieszczonych w
* zmiennej opcje
```

```
DO WHILE x1 < x2
        @ x1,y1 PROMPT SUBSTR ( opcje,s,w)
        x1 = x1 + 1
        s = s + w
ENDDO
@ x,y TO x2,y2 DOUBLE
SET KEY zn TO Znak && naciśniecie klawisza ze znakiem o
                   && kodzie umieszczonym w zmiennej zn
                   && powoduje wpisanie do bufora
                   && klawiatury znaku powrotu
MENU TO kontr
RETURN
* Procedura zapamiętywania faktu naciśnięcia odpowiedniego
* klawisza
PROCEDURE Znak
PARAMETERS p1,p2,p3
KEYBOARD chr(13)
wyn = .T.
RETURN
```

Procedura Menu umożliwia wyświetlenie prostego menu systemu, w którym opcje podawane są w postaci tekstu umieszczonego w jednej zmiennej. Ponadto można w sposób dowolny zdefiniować znak, który spowoduje wybranie danej opcji. Można też otrzymać informację, czy wybranie danej opcji nastąpiło w sposób standardowy, czy też na skutek naciśnięcia wyróżnionego znaku. Przykład wywołania procedury Menu zamieszczono poniżej.

Przykład 8.

```
CLEAR
wynik = .F.
nr = 0
klawisz = 22
DO Menu WITH 10,10,15,3,;
' pierwsza opcja druga opcja     trzecia opcja ', nr,klawisz,wynik
? nr,wynik
WAIT " "
RETURN
```

W powyższym wywołaniu procedury Menu wyróżnionym klawiszem jest klawisz o kodzie 22 (jest to klawisz INS). Naciśnięcie go spowoduje wybranie odpowiedniej opcji z menu i jednocześnie zmienna wynik otrzyma wartość .

Na zakończenie przedstawiony zostanie przykład wykorzystania funkcji **ACHOICE** dostępnej w bibliotece EXTEND.LIB.

Przykład 9.

```
*************************************************************
*       program ilustrujacy wywołanie funkcji ACHOICE       *
*       tworzącej menu systemu                              *
*************************************************************
CLEAR
DECLARE tab[3]
tab[1] = 'pierwsza opcja'
tab[2] = 'druga opcja'
tab[3] = 'trzecia opcja'
s = ACHOICE (0,0,2,14,tab)
@ 10,0 SAY s
WAIT "
RETURN
```

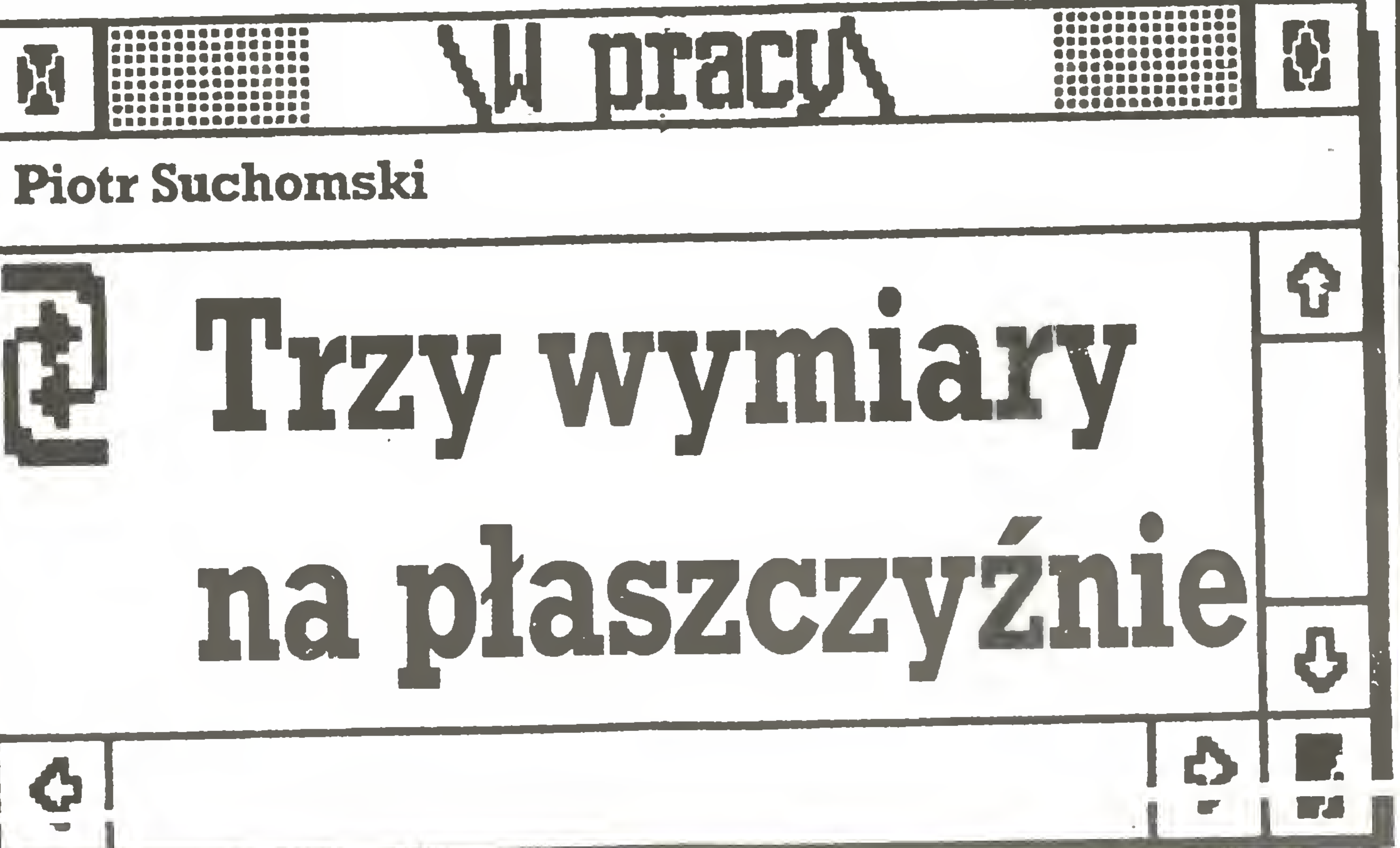
Parametrami funkcji ACHOICE wywołanej w przykładzie są współrzędne lewego górnego rogu i prawego dolnego rogu ramki, w której wyświetlone będzie menu. Tablica tab zawiera opcje menu, każda opcja umieszczona jest w poszczególnym elemencie tablicy.

Przedstawione w artykule programy tworzenia menu systemu zapisane w wersji źródłowej na dyskietce oraz opis języka Clipper w postaci programu rezydującego można zamówić listownie u autora artykułu (adres i telefon w ogłoszeniu w numerze 3/89 "Komputera").

LITERATURA

1. Baker R.H.: Advanced dBase III applications. Tab Books Ins. PA17214.
2. Date C.J.: Wprowadzenie do baz danych. WNT Warszawa 1981.
3. dBase III PLUS, User Manual. Ashton-Tate.
4. Clipper, User Manual. Nantucket.
5. Jones E.:dBase III PLUS. Power user's guide. Osborne McGraw Hill, Berkeley, California. 1988.
6. Salzberg B.J.: An introduction to data base design. Academic Press College Division 1986.
7. Ullman J.D.: Systemy baz danych. WNT Warszawa 1988.

W artykule wykorzystano fragmenty książki: Krzysztof Walczak "Programowanie systemów baz danych. Język Clipper", przygotowywanej do druku w WNT.



Żaden naprawdę płaski, dwuwymiarowy obraz nie istnieje.
Rudolf Arnheim : "Sztuka i percepcja wzrokowa"

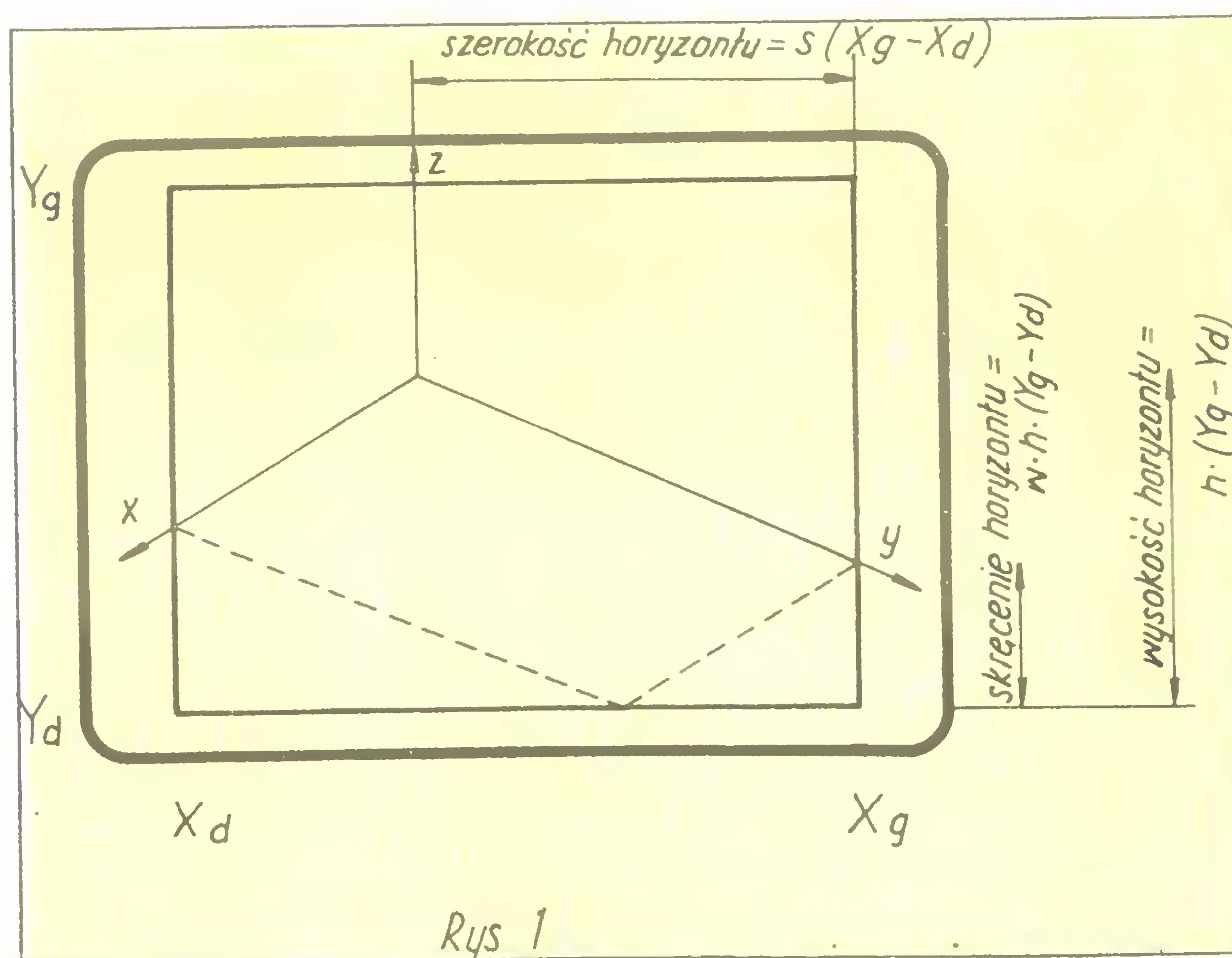
Jednym z interesujących problemów grafiki komputerowej jest adekwatne odwzorowanie na płaszczyźnie powierzchni trójwymiarowej, będącej zadaną funkcją dwóch zmiennych. Graficzna reprezentacja zmienności takich funkcji jest bowiem bezsprzecznie najbardziej poglądową metodą ich badania.

Problem ten wymaga rozwiązania dwóch zagadnień szczegółowych. Pierwsze z nich dotyczy sposobów rzutowania punktów przestrzeni trójwymiarowej na płaszczyznę – obraz badanej funkcji otrzymuje się bowiem na podstawie rzutów wartości tej funkcji dla wybranych dyskretnych punktów jej dziedziny, tworzących zwykle węzły sieci prostokątnej. Najczęściej wykorzystuje się tu rzut aksonometryczny – czyli rzut, który nie zmienia miary odległości wzdłuż rzutowanego kierunku – co sprowadza się do znanych i nieskomplikowanych transformacji układów współrzędnych [1]-[3]. Tak otrzymane punkty na płaszczyźnie rzutowania łączy się następnie odcinkami linii prostych, przyjmując, że są one odwzorowaniem wartości badanej funkcji na odcinkach między sąsiednimi węzłami sieci.

Niestety, każdy, kto próbował swoich sił, dobrze pamięta jak rozczarowujący był efekt zastosowania powyższego przepisu. Aby otrzymany tą drogą obraz uczynić dostatecznie czytelnym, konieczne jest jeszcze bowiem rozwiązanie kolejnego i znacznie trudniejszego zagadnienia. Polega ono na eliminowaniu tych fragmentów obrazu, które są w rzeczywistości niewidoczne. W praktyce – ze względu na wymagania pamięciowe oraz czasowe – znany ogólny algorytm tak zwanej linii zasłoniętej [1] znajduje tu tylko ograniczone zastosowanie. Ciągle aktualne jest zatem poszukiwanie algorytmów alternatywnych – uproszczonych i w konsekwencji bardziej efektywnych. Taki przybliżony algorytm linii zasłoniętej powinien jednak charakteryzować się dostateczną uniwersalnością oraz jakością wizualną odwzorowanej powierzchni.

Przejdźmy zatem do omówienia algorytmu proponowanego w

> 34



Rys 1

33 <

artykule, spełniającego – zdaniem autora – postawione wyżej wymagania.

Założenia algorytmu

1. Zakładamy, że pole rzutowania utworzone jest na ekranie przez prostokąt określony parami współrzędnych (*Xd, Yg*) i (*Xg, Yd*) swoich wierzchołków (rys. 1).

2. Obrazy osi *X, Y, Z* rzutowanego ortogonalnego układu współrzędnych wyznaczone są – przy założeniu, że oś *Z* jest równoległa do osi *Y* ekranu – przez przyjęcie następujących trzech wielkości: szerokości, wysokości oraz skręcenia horyzontu (rys.1). Wygodnie jest przy tym operować względnymi miarami tych wielkości – to znaczy liczbami z przedziału <0,1> – oznaczonymi odpowiednio *s,h,* oraz *w* i także zdefiniowanymi na powyższym rysunku.

3. Wprowadzone parametry rzutowania *s,h,w* stanowią stopnie swobody omawianego algorytmu, pozwalające na zmianę "punktu widzenia", w tym także na poszukiwanie takiego ujęcia obrazu rzutowanej powierzchni trójwymiarowej $z = f(x,y)$, które zapewni maksymalną jego czytelność. W szczególności – jak łatwo zauważyć – gdy $w = 0$, płaszczyzna *xz* rzutowanego układu współrzędnych pozostaje równoległa do płaszczyzny ekranu, zaś dla $w = 1$, płaszczyzną taką jest płaszczyzna *yz*.

4. W prezentowanym algorytmie bezpośrednio dobierane są tylko dwa parametry rzutowania: wysokość i skręcenie horyzontu. Szerokość zaś określana jest w sposób pośredni, zawsze jednak tak, aby obrazy węzłów rzutowanej płaszczyzny *xy* charakteryzowały się pewnymi narzuconymi uporządkowaniami, co w konsekwencji – jak pokażemy – pozwala na sformułowanie szczególnie prostej zasady linii zasłoniętej. Proponuje się zatem wprowadzenie następującej klasyfikacji owych uporządkowań, wyróżniając ich indeks $k = 1,2,...$:

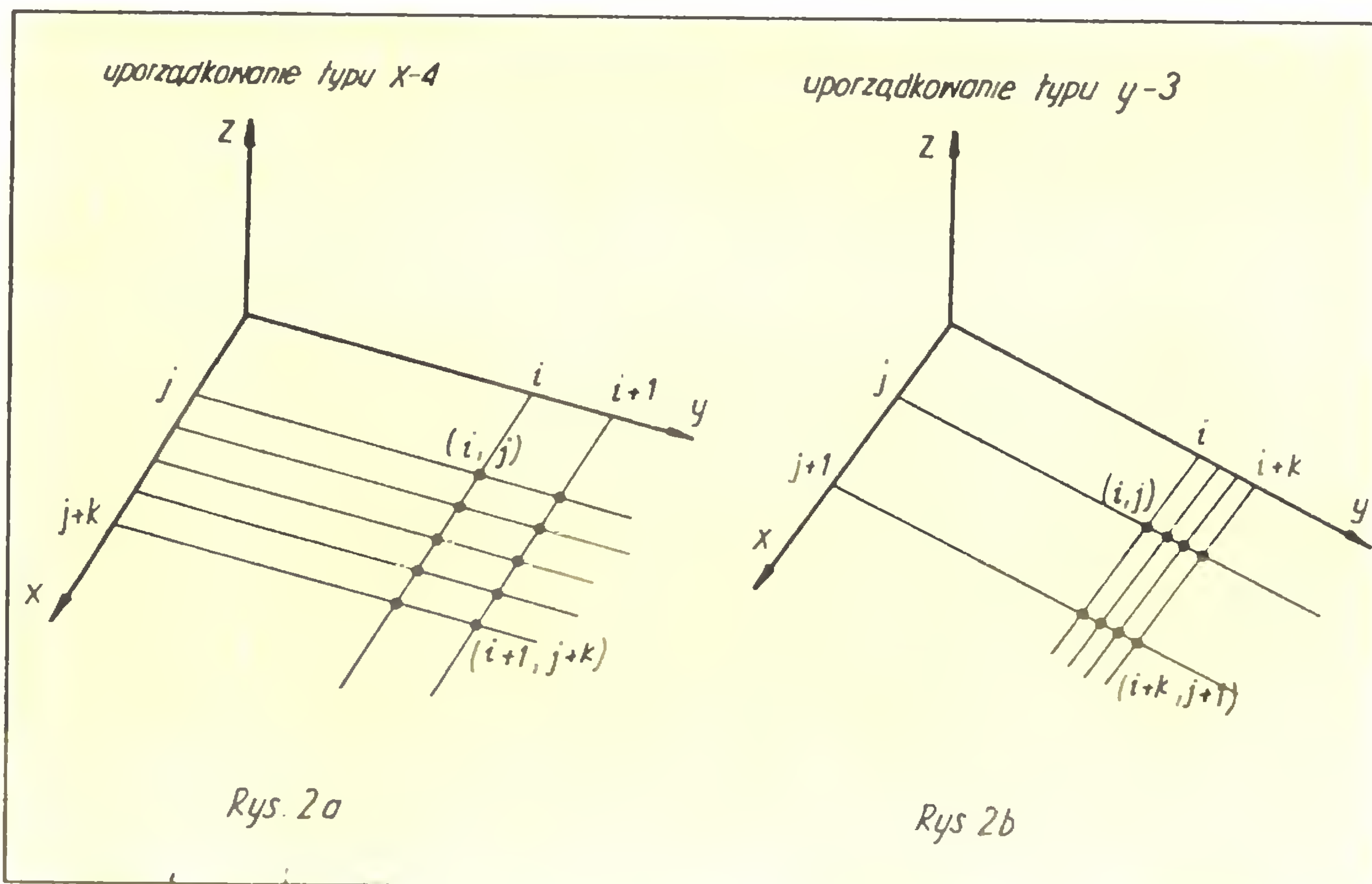
– uporządkowanie typu *x-k*: $X[i,j] = X[i+1,j+k]$, (rys.2a);
– uporządkowanie typu *y-k*: $X[i,j] = X[i+k,j+1]$, (rys.2b).

Wskaźniki *i* oraz *j*, występujące w powyższym zapisie, odnoszą się do kolejnych węzłów sieci uzyskanej przez równomierny podział osi rzutowanego układu współrzędnych – odpowiednio *y* oraz *x*.

Jak łatwo zauważyć, przy ustalonych podziałach *nx* oraz *ny* osi rzutowanego układu współrzędnych, decydując o typie uporządkowania, określa się tym samym szerokość horyzontu. Dla uporządkowania typu *x-k* mamy $s = k*ny/(nx+k*ny)$, zaś w przypadku uporządkowania typu *y-k* zachodzi $s = ny/(ny+k*nx)$. Zauważmy, że zwiększanie indeksu *k* w uporządkowaniu typu *x-k* prowadzi do poszerzania horyzontu, zaś w uporządkowaniu typu *y-k* do jego zawężania.

Opis algorytmu

Opierając się na poczynionych założeniach, można sformułować następujący dwukrokowy algorytm płaskiej reprezentacji powierzchni trójwymiarowej – w opisie pominięto jego wstępne kroki przygotowawcze, których szczegóły można łatwo odczytać z załączonego tekstu źródłowego w Turbo Pascalu.



Rys. 2a   Rys. 2b

Krok 1. Dla wskaźników *i,j* zmieniających się w sposób malejący, dokonuje się modyfikacji współrzędnych *Y* ekranu rzutów wartości funkcji w tych węzłach (*i,j*), dla których są one zasłonięte przez inne elementy tworzonego obrazu. Modyfikacja taka polega na zastąpieniu bieżącej wartości współrzędnej *Y* maksymalną z wartości zasłaniających – co oczywiście nie zakłóca postrzegania widzialnych elementów obrazu. Przykładowo, dla uporządkowania typu *x-k* modyfikacja powyższa (rys. 2a) wymaga tylko k-krotnego porównania *Y[i,j]* z wartościami tej współrzędnej punktów leżących na prostej łączącej rzuty wartości funkcji w węzłach (*i,j+1*)

```
Program Grafika_przestrzenna_720X350;

 {$I typedef.sys }
 {$I graphix.sys }
 {$I kernel.sys  }

type  tablica = array [0..40,0..40] of real;
var
 i,j,nx,ny,k       : integer;
 h,w,z_min,z_max   : real;
 typ               : char;
 Z               . : tablica;

 procedure grafika
  ( nx,ny,k                : integer;
    h,w,z_min,z_max        : real;
    typ                    : char;
    Z                      : tablica);
  {
    UWAGI :
    1. Tablica  Z  zawiera  wartosci ( typu real )  przedstawianej
       funkcji dwoch zmiennych z = f(x,y) w punktach x(j) i y(i) :
          Z[i,j]  -  f(x(j),y(i)) ; i = 0,...,ny, j = 0,...,nx.
    2. Wartosci parametrow z_min i z_max definiuja zakres osi z na
       wykresie funkcji f(x,y).
    3. Wymiary pola wykresu  odpowiadaja rozdzielczosci 720 X 350.
    4. Zastosowano procedury biblioteczne z pakietu Turbo Graphix.
  }
  var
   i,j,l,x_tu,y_tu,xd,xg,yd,yg,xh,yh : integer;
   s,dxi,dxj,dyi,dyj,dz              : real;
   X,Y  : array [0..40,0..40] of integer;

  procedure krok ( x_tam,y_tam : integer );
   begin
    DrawLine (x_tu,y_tu,x_tam,y_tam);
    x_tu := x_tam; y_tu := y_tam;
   end; (* krok *)

  function maks ( i,j,m : integer; typ : char ) : integer;
   var  l,max,y1 : integer;
   begin
    max := yd;
    for l := m downto 0 do
     case typ of
      'x' : if j+l <= nx then
             begin
              y1 := Y[i+1,j+l]-round((m-l)*(Y[i+1,j+l]-Y[i,j+l])/k);
              if max > y1 then max := y1
             end;
      'y' : if i+l <= ny then
             begin
              y1 := Y[i+l,j+1]-round((m-l)*(Y[i+l,j+1]-Y[i+l,j])/k);
              if max > y1 then max := y1
             end
     end;
    maks := max;
   end; (* maks *)
 procedure korekcja_Y ( n1,n2 : integer; typ : char );
  var i,j,l,y1 : integer;
  begin
   for i := n1-1 downto 0 do
    for j := n2-1 downto 0 do
     for l := j+k downto j+1 do
      if l <= n2 then
       case typ of
        'x' : begin
               y1 := round(Y[i,l] - (l-j)*(Y[i,l]-Y[i+1,l])/k);
               if Y[i,j] > y1 then Y[i,j] := y1
              end;
        'y' : begin
               y1 := round (Y[l,i] - (l-j)*(Y[l,i]-Y[l,i+1])/k);
               if Y[j,i] > y1 then Y[j,i] := y1
              end
       end
  end; (* korekcja Y *)

begin  (* czesc operacyjna procedury grafika *)
 InitGraphic;ClearScreen;
 DefineWorld (1,0,0,720,350); SelectWorld (1);
 xd := 37; yd := 340; xg := 682; yg := 10;
 case typ of
  'x' : s := k*ny/(nx+k*ny);
  'y' : s :=  ny /(ny+k*nx);
 end;
 xh  := xd+round((1-s)*(xg-xd)); yh := yd-round(h*(yd-yg));
 dxi := s*(xg-xd)/ny; dxj := (1-s)*(xg-xd)/nx;
 dyi := (1-w)*h*(yd-yg)/ny; dyj := w*h*(yd-yg)/nx;
 dz  := (1-h)*(yd-yg)/(z_max-z_min);
 for i := 0 to ny do
  for j := 0 to nx do
   begin
    X[i,j] := xh + round(i*dxi-j*dxj);
    Y[i,j] := yh + round(i*dyi+j*dyj-dz*(Z[i,j]-z_min));
   end;
 case typ of
  'x' : begin
         korekcja_Y (ny,nx,'x');
         for i := 0 to ny-1 do
          begin
           for j := 0 to nx-1 do
            begin
             x_tu := X[i+1,j]; y_tu := Y[i+1,j];
             for l := 1 to k-1 do
              krok (round(x_tu-dxj),maks(i,j,l,'x'));
             krok (X[i,j],Y[i,j]); krok (X[i,j+1],Y[i,j+1]);
            end;
           krok (X[i+1,nx],Y[i+1,nx]);
          end;
         for j := 0 to nx-1 do
          DrawLine (X[ny,j],Y[ny,j],X[ny,j+1],Y[ny,j+1]);
        end;
  'y' : begin
         korekcja_Y (nx,ny,'y');
         for j := 0 to nx-1 do
```

```
            begin
             for i := 0 to ny-1 do
              begin
               x_tu := X[i,j+1]; y_tu := Y[i,j+1];
               for l := 1 to k-1 do
                krok (round(x_tu+dxi),maks(i,j,l,'y'));
               krok (X[i,j],Y[i,j]); krok (X[i+1,j],Y[i+1,j]);
              end;
             krok (X[ny,j+1],Y[ny,j+1]);
            end;
           for i := 0 to ny-1 do
            DrawLine (X[i,nx],Y[i,nx],X[i+1,nx],Y[i+1,nx]);
          end
   end;
   repeat until keypressed;
   LeaveGraphic;
  end; (* grafika *)

begin  (* przyklad *)
 nx := 20; ny := 20; h := 0.65; w := 0.75; k := 5; typ := 'x';
 z_min := -3; z_max := 3;
 clrscr; writeln ('OBLICZANIE FUNKCJI Z = F(X,Y) ...');
 for i := 0 to ny do
  for j := 0 to nx do
   Z[i,j] := 3*sin(sqrt(sqr(j-10)+sqr(i-10)));
 grafika (nx,ny,k,h,w,z_min,z_max,typ,Z);
end.
```

```
 {$I typedef.sys }
 {$I graphix.sys }
 {$I kernel.sys  }

type  tablica = array [0..40,0..40] of real;
var
 i,j,nx,ny,k      : integer;
 h,w,z_min,z_max : real;
 typ              : char;
 Z                : tablica;

 procedure grafika
  ( nx,ny,k              : integer;
    h,w,z_min,z_max      : real;
    typ                  : char;
    Z                    : tablica);
  {
    UWAGI :
    1. Tablica  Z  zawiera  wartosci ( typu real )  przedstawianej
       funkcji dwoch zmiennych z = f(x,y) w punktach x(j) i y(i) :
          Z[i,j]  =  f(x(j),y(i)) ; i = 0,...,ny, j = 0,...,nx.
    2. Wartosci parametrow z_min i z_max definiuja zakres osi z na
       wykresie funkcji f(x,y).
    3. Wymiary pola wykresu  odpowiadaja rozdzielczosci 640 X 200.
    4. Zastosowano procedury biblioteczne z pakietu Turbo Graphix.
  }
  var
   i,j,l,x_tu,y_tu,xd,xg,yd,yg,xh,yh : integer;
   s,dxi,dxj,dyi,dyj,dz              : real;
   X,Y  : array [0..40,0..40] of integer;

  procedure krok ( x_tam,y_tam : integer );
   begin
    DrawLine (x_tu,y_tu,x_tam,y_tam);
    x_tu := x_tam; y_tu := y_tam;
   end; (* krok *)

  function maks ( i,j,m : integer; typ : char ) : integer;
   var  l,max,y1 : integer;
   begin
    max := yd;
    for l := m downto 0 do
     case typ of
      'x' : if j+l <= nx then
             begin
              y1 := Y[i+1,j+l]-round((m-l)*(Y[i+1,j+l]-Y[i,j+l])/k);
              if max < y1 then max := y1
             end;
      'y' : if i+l <= ny then
             begin
              y1 := Y[i+l,j+1]-round((m-l)*(Y[i+l,j+1]-Y[i+l,j])/k);
              if max < y1 then max := y1
             end
     end;
    maks := max;
   end; (* maks *)

  procedure korekcja_Y ( n1,n2 : integer; typ : char );
   var i,j,l,y1 : integer;
   begin
    for i := n1-1 downto 0 do
     for j := n2-1 downto 0 do
      for l := j+k downto j+1 do
       if l <= n2 then
        case typ of
         'x' : begin
                y1 := round(Y[i,l] - (l-j)*(Y[i,l]-Y[i+1,l])/k);
                if Y[i,j] < y1 then Y[i,j] := y1
               end;
         'y' : begin
                y1 := round (Y[l,i] - (l-j)*(Y[l,i]-Y[l,i+1])/k);
                if Y[j,i] < y1 then Y[j,i] := y1
               end
        end
   end; (* korekcja Y *)

 begin  (* czesc operacyjna procedury grafika *)
  InitGraphic;ClearScreen;
  DefineWorld (1,0,0,640,200); SelectWorld (1);
  xd := 37; yd := 10; xg := 602; yg := 190;
  case typ of
   'x' : s := k*ny/(nx+k*ny);
   'y' : s :=  ny /(ny+k*nx);
  end;
  xh  := xd+round((1-s)*(xg-xd)); yh := yd-round(h*(yd yg));
  dxi := s*(xg-xd)/ny; dxj := (1 s)*(xg-xd)/nx;
  dyi := (1 w)*h*(yd-yg)/ny; dyj := w*h*(yd-yg)/nx;
  dz  := (1-h)*(yd-yg)/(z_max-z_min);
  for i := 0 to ny do
   for j := 0 to nx do
    begin
     X[i,j] := xh + round(i*dxi-j*dxj);
     Y[i,j] := yh + round(i*dyi+j*dyj dz*(Z[i,j] z_min));
    end;
    case typ of
     'x' : begin
            korekcja_Y (ny,nx,'x');
            for i := 0 to ny-1 do
             begin
              for j := 0 to nx-1 do
               begin
                x_tu := X[i+1,j]; y_tu := Y[i+1,j];
                for l := 1 to k-1 do
                 krok (round(x_tu-dxj),maks(i,j,l,'x'));
                krok (X[i,j],Y[i,j]); krok (X[i,j+1],Y[i,j+1]);
               end;
              krok (X[i+1,nx],Y[i+1,nx]);
             end;
            for j := 0 to nx-1 do
             DrawLine (X[ny,j],Y[ny,j],X[ny,j+1],Y[ny,j+1]);
           end;
     'y' : begin
            korekcja Y (nx,ny,'y');
             for j := 0 to nx-1 do
              begin
               for i := 0 to ny-1 do
                begin
                 x_tu := X[i,j+1]; y_tu := Y[i,j+1];
                 for l := 1 to k-1 do
                  krok (round(x_tu+dxi),maks(i,j,l,'y'));
                 krok (X[i,j],Y[i,j]); krok (X[i+1,j],Y[i+1,j]);
                end;
               krok (X[ny,j+1],Y[ny,j+1]);
              end;
             for i := 0 to ny-1 do
              DrawLine (X[i,nx],Y[i,nx],X[i+1,nx],Y[i+1,nx]);
            end
    end;
    repeat until keypressed;
    LeaveGraphic;
   end; (* grafika *)

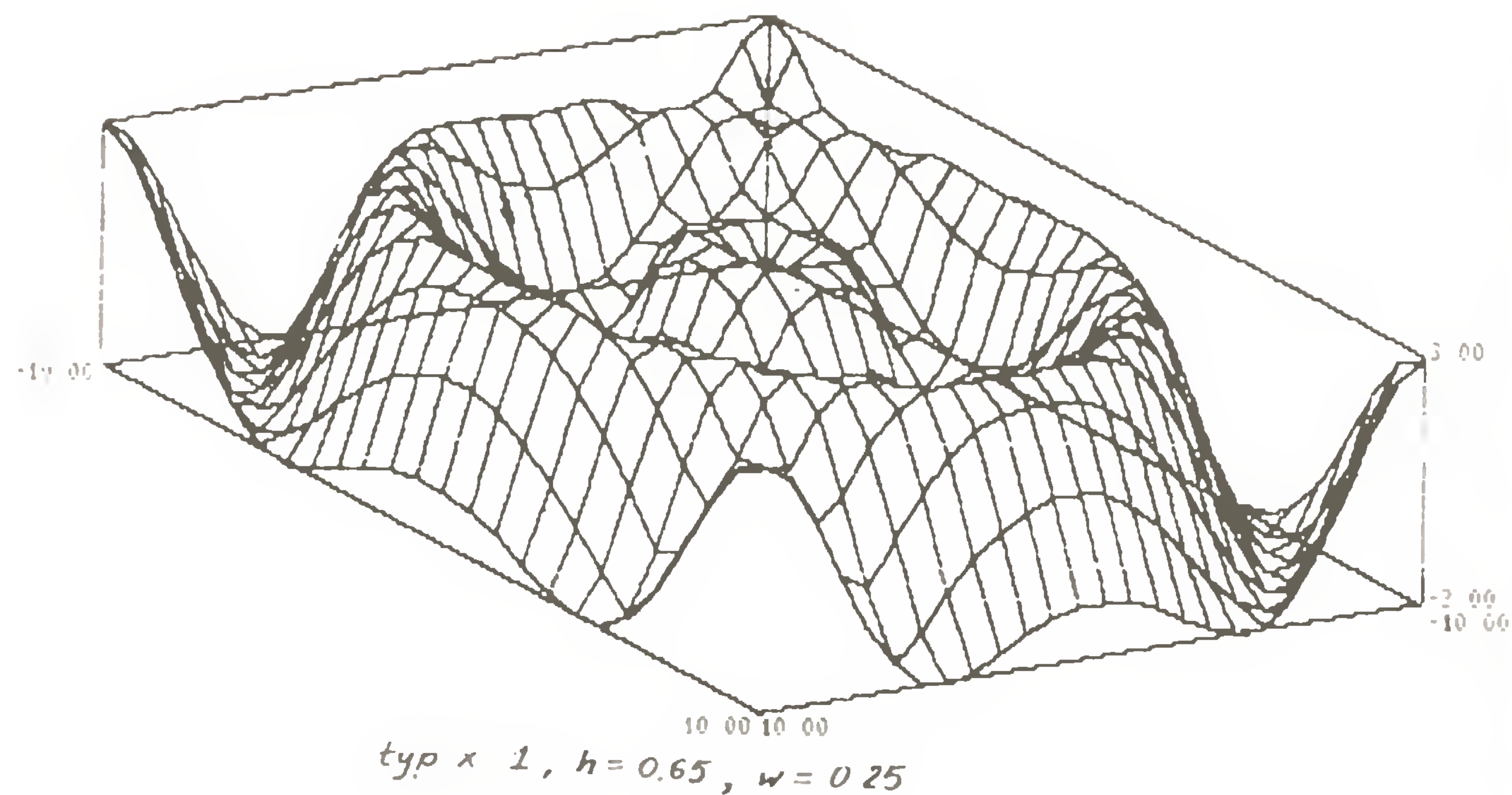
begin  (* przyklad *)
 nx := 20; ny := 20; h := 0.65; w := 0.75; k := 5; typ := 'x';
 z_min := -3; z_max := 3;
 clrscr; writeln ('OBLICZANIE FUNKCJI Z = F(X,Y) ...');
 for i := 0 to ny do
  for j := 0 to nx do
   Z[i,j] := 3*sin(sqrt(sqr(j-10)+sqr(i-10)));
 grafika (nx,ny,k,h,w,z_min,z_max,typ,Z);
end.
```

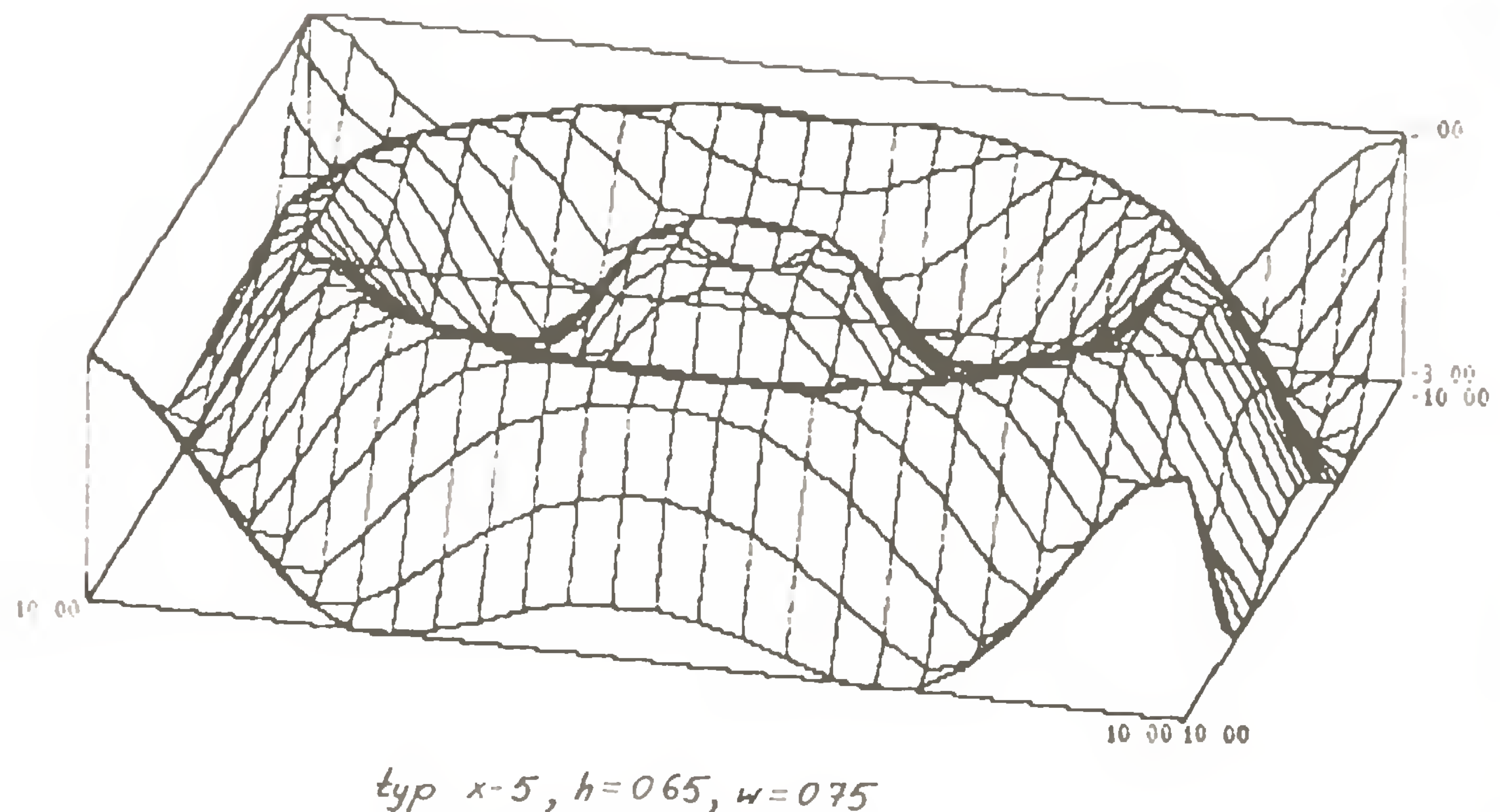
oraz $(i+1,j+l)$, dla $l=1,...,k$, przy ustalonej współrzędnej $X = X[i,j]$.

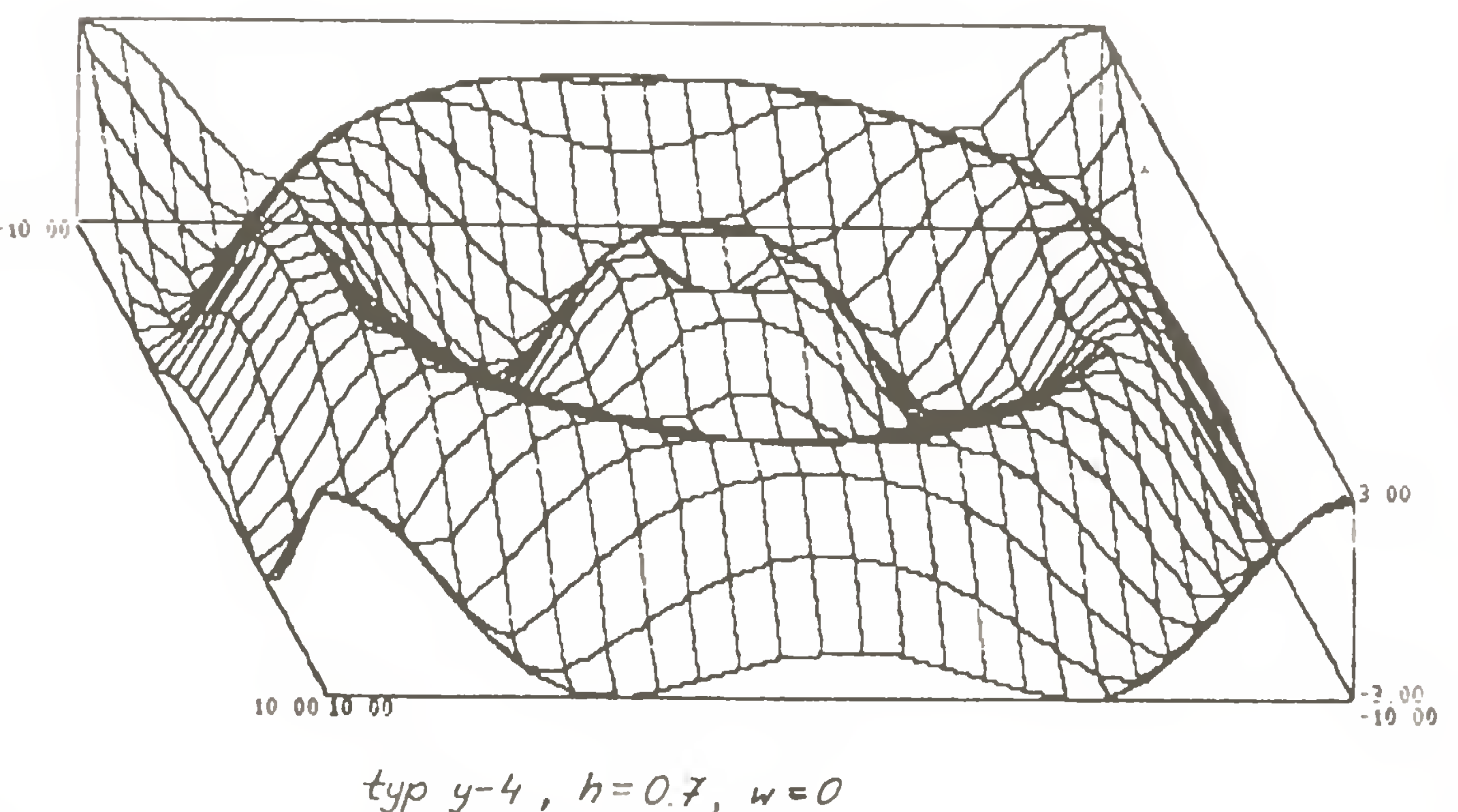
Krok 2. Dla wskaźników i,j zmieniających się w sposób rosnący, łączy się punkt odpowiadający rzutowi wartości rozważanej funkcji w węźle (i,j) z rzutami odpowiadającymi węzłom sąsiednim – to znaczy węzłom $(i,j+1)$ oraz $(i+1,j)$. Sposób owego połączenia omówimy znowu na przykładzie uporządkowania typu *x-k*. Wobec modyfikacji dokonanej w kroku 1, połączenie z punktem odpowiadającym wartości funkcji w węźle $(i,j+1)$ wykonywane jest bezpośrednio za pomocą odcinka linii prostej – oba łączone punkty są bowiem widoczne i linia je łącząca nie może być niczym zasłonięta. Połączenie z punktem odpowiadającym węzłowi $(i+1,j)$ w ogólności przebiega według linii łamanej złożonej z k odcinków – musi bowiem uwzględniać ewentualne zasłonięcia prostej łączącej w maksymalnie k-1 punktach. Współrzędne X ekranu końców kolejnych odcinków tej łamanej odpowiadają współrzędnym X węzłów leżących między węzłem $(i+1,j)$ a węzłem $(i+1,j+k)$. Zaś jako współrzędną *Y* ekranu końca kolejnego odcinka konstruowanej łamanej przyjmuje się bądź wartość bieżącą współrzędnej *Y* prostej łączącej rzuty wartości funkcji w węzłach (i,j) i $(i+1,j)$, bądź też – w przypadku jej zasłaniania – maksymalną z wartości zasłaniających. Wobec uczynionych założeń o typie uporządkowania węzłów, wartości współrzędnych *Y* punktów potrzebnych do takich porównań oblicza się bardzo łatwo – dla danego odcinka łamanej są to bowiem punkty o ustalonej i określonej wyżej wartości współrzędnej *X*, przy czym ich liczba jest oczywiście skończona i równa się maksymalnie $k+1$.

W przypadku uporządkowania typu *y-k* postępowanie jest analogiczne – w sposób bezpośredni łączy się teraz punkty odpowiadające rzutom wartości funkcji w węzłach (i,j) oraz $(i+1,j)$, zaś przy pomocy łamanej – w węzłach (i,j) oraz $(i,j+1)$.

> 36







Wyniki zastosowania proponowanego przepisu Czytelnik najlepiej oceni sam na podstawie rys. 3. Przedstawione tam wykresy przykładowej funkcji $f = 3sinrx*x + y*y$, otrzymano dla różnych typów uporządkowania węzłów. Interesującą kwestią pozostaje wybór typu uporządkowania, a także wartości parametrów rzutowania *h* i *w* – tak, by odpowiadający im obraz charakteryzował się możliwie dużą czytelnością. Trudno tu oczekiwać ścisłych rozwiązań, zachęcamy jednak do poszukiwania reguł heurystycznych, pozwalających – być może - na algorytmizację owego wyboru.

Kończąc, warto jeszcze zauważyć, że omówiony algorytm tylko w sposób przybliżony realizuje zasadę linii zasłoniętej – np. linie na tak otrzymanym obrazie mogą stykać się wyłącznie w punktach o współrzędnych *X* odpowiadających przyjętym węzłom sieci. Jak się jednak wydaje, nie musi to w sposób istotny wpływać na pogorszenie jakości obrazu, radykalnie zaś przyspiesza jego otrzymanie (ze wzrostem indeksu *k* szybkość ta maleje). Wprowadzony podział typów uporządkowania węzłów sieci ma charakter umowny, może wszakże okazać się przydatny przy klasyfikowaniu i porównywaniu różnych algorytmów proponowanych przez innych autorów. Przykładowo, w pracach [1] i [4] oparto się na bardzo prostym uporządkowaniu typu *x-1*, zaś procedury numeryczne prezentowane w pracy [5] dotyczą m.in. uporządkowania typu *y-2*.

Literatura

[1] Angell I. O. : Wprowadzenie do grafiki komputerowej. WNT, Warszawa, 1986.
[2] Pavlidis T. : Grafika i przetwarzanie obrazów. WNT, Warszawa, 1987.
[3] Hearn D., Baker M. P. : Microcomputer Graphics. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.
[4] Gawrikow R. : 3D-Netzgrafik mit Turbo Pascal. Computer Personlich, 23/86, 68-69.
[5] Cuellar G. : Graphismes sur IBM PC. Eyrolles, Paris, 1986.



Pod tym tytułem otwieramy rubrykę poświęconą dostosowywaniu do potrzeb użytkownika, oswajaniu gotowych, istniejących już programów. Nowoczesne programy mikrokomputerowe zawierają w paletach swoich komend i opcji jedną szczególnie ciekawą zdolność nazywaną z reguły w instrukcjach ich użytkowania – customizing. To dzięki niej można np. w programie 1-2-3 zaprogramować menu całkowicie w polskim języku (bez ingerowania w sam program) czy zmienić znak zachęty w interpreterze dBase. Sztuka odpowiedniego zaprogramowania plików wsadowych (tzw. batchy) to także oswajanie programu, w tym przypadku systemu operacyjnego.

Wprowadzenie swoich opcji i poleceń jest zdolnością bardzo cenną. Tak, jak idea mikrokomputera IBM PC pozwala na elastyczne dostosowanie konfiguracji sprzętu do potrzeb użytkownika, tak właśnie zmiany w menu programu dostosowują go do potrzeb tego, który z nim pracuje.

W pierwszej kolejności proponujemy zająć się programem AutoCAD. W tym odcinku proponujemy dołączenie nowej opcji do menu programu. W następnym numerze podamy nieco szczegółów dotyczących użycia wewnętrznego języka AutoCAD-a - AutoLispa.

Redakcja

Coś nowego w MENU programu AutoCAD

Jedną z metod "oswojenia" czyli przystosowania AutoCAD-a do własnych potrzeb jest możliwość dokonywania zmian w menu ekranowym.

Na przykład: dodajmy do głównego menu opcję "Licz". Po jej wybraniu pojawi się menu pozwalające wykonać proste obliczenia, takie jak DODAWANIE, ODEJMOWANIE, MNOŻENIE, DZIELENIE, POTĘGOWANIE, SINUS, COSINUS, TANGENS.

Przystąpmy zatem do modyfikacji menu. Do dowolnego edytora wczytujemy plik ACAD.MNU i między opcje [UTILITY] i [ASHADE] wpisujemy naszą opcję [Licz]$S=X $S=licz (patrz niżej). Następnie przed menu o nazwie **BL 3 wpisujemy całą zawartość naszego menu **licz 3 (patrz poniżej). Zapisujemy na dysku poprawiony plik ACAD.MNU i uruchamiamy AUTOCAD-a. Po wejściu do edytora graficznego na ekranie pojawi się poprawione menu – sprawdźmy jak działa nasz kalkulator. Wybieramy opcję "licz", a następnie opcję "podaj liczbę" i na żądanie "podaj liczbę ---- >" wpiszmy "2", następnie wybieramy "Dodaj" i na żądanie programu podajemy "3" – powinien pojawić się wynik 5, możemy teraz wybrać "Mnoz" i podać "6" a pojawi się wynik 30, w ten sam sposób liczyć możemy dalej.

Wpisanie "!odp" w dowolnej chwili np. w odpowiedzi na pytanie o promień okręgu spowoduje przyjęcie (jako promień) ostatnio policzonej wartości, w naszym przykładzie – liczby 30.

```
***SCREEN
**S
[AUTOCAD]$S=S
[* * * *]$S=OSNAPB
[SETUP]^C^C(load "setup") $S=UNITS
[BLOCKS]$S=X $S=BL
[DIM:]$S=X $S=DIM ^C^CDIM
[DISPLAY]$S=X $S=DS
[DRAW]$S=X $S=DR
[EDIT]$S=X $S=ED
[INQUIRY]$S=X $S=INQ
```

```
[LAYER:]$S=X $S=LAYER ^C^CLAYER
[SETTINGS]$S=X $S=SET
[PLOT]$S=X $S=PLOT
[UTILITY]$S=X $S=UT
[Licz ]$S=X $S=licz
[ASHADE ]$S=X $S=ASHADE (IF (NULL AS)(PROGN (VMON)(SETQ AS 1)(LOAD "ASHADE")))
[3D]$S=X $S=3D
[SAVE:]^C^CSAVE
**LICZ 3
[sinus]^C^C(setq odp (sin (* odp 0.01745329)));
[cosinus]^C^C(setq odp (cos (* odp 0.01745329)));
[tan]^C^C(setq t1 (sin (* odp 0.01745329)))+
(setq t2 (cos (* odp 0.01745329)))+
(setq odp (/ t1 t2))
[Dziel]^C^C(setq odp (/ (getreal "podaj przez co podzielic ---> ")));
[Mnoz]^C^C(setq odp (* (getreal "podaj przez co pomnozyc ---> ")));
[Dodaj]^C^C(setq odp (+ (getreal "podaj co dodac ---> ")));
[Odejmij]^C^C(setq odp (- (getreal "podaj co odjac ---> ")));
[Kwadrat]^C^C(setq odp (expt odp 2))
[Pierwias]^C^C(setq odp (sqrt odp))
[Potega]^C^C(setq odp (expt odp (getreal "podaj wykladnik ---> ")));
[Podaj]^C^C(setq odp (getreal "podaj liczbe ---> "));
[liczbe]^C^C(setq odp (getreal "podaj liczbe ---> "));
**BL 3
[ATTDEF:]$S=X $S=ATTDEF ^C^CATTDEF
[BASE:]$S=X $S=BASE ^C^CBASE
[BLOCK:]$S=X $S=BLOCK ^C^CBLOCK
[INSERT:]$S=X $S=INSERT ^C^CINSERT
[MINSERT:]$S=X $S=MINSERT ^C^CMINSERT
[WBLOCK:]$S=X $S=WBLOCK ^C^CWBLOCK
```

Pomysł z "CAD User" 4/88 "ulepszył" ***Zbigniew Blewoński***



W poprzednich odcinkach przedstawiliśmy drogę rozwoju drukarstwa od tajemniczych znaków rytych w kamieniu do użycia pierwszych komputerów. Obecnie, za sprawą bardzo szybkiego rozwoju elektroniki i systemów komputerowych, składanie tekstów oraz ich druk stał się ogólnie dostępny. Nawet najmniejsze komputery domowe są w stanie podołać procesom obróbki tekstów i ich druku.

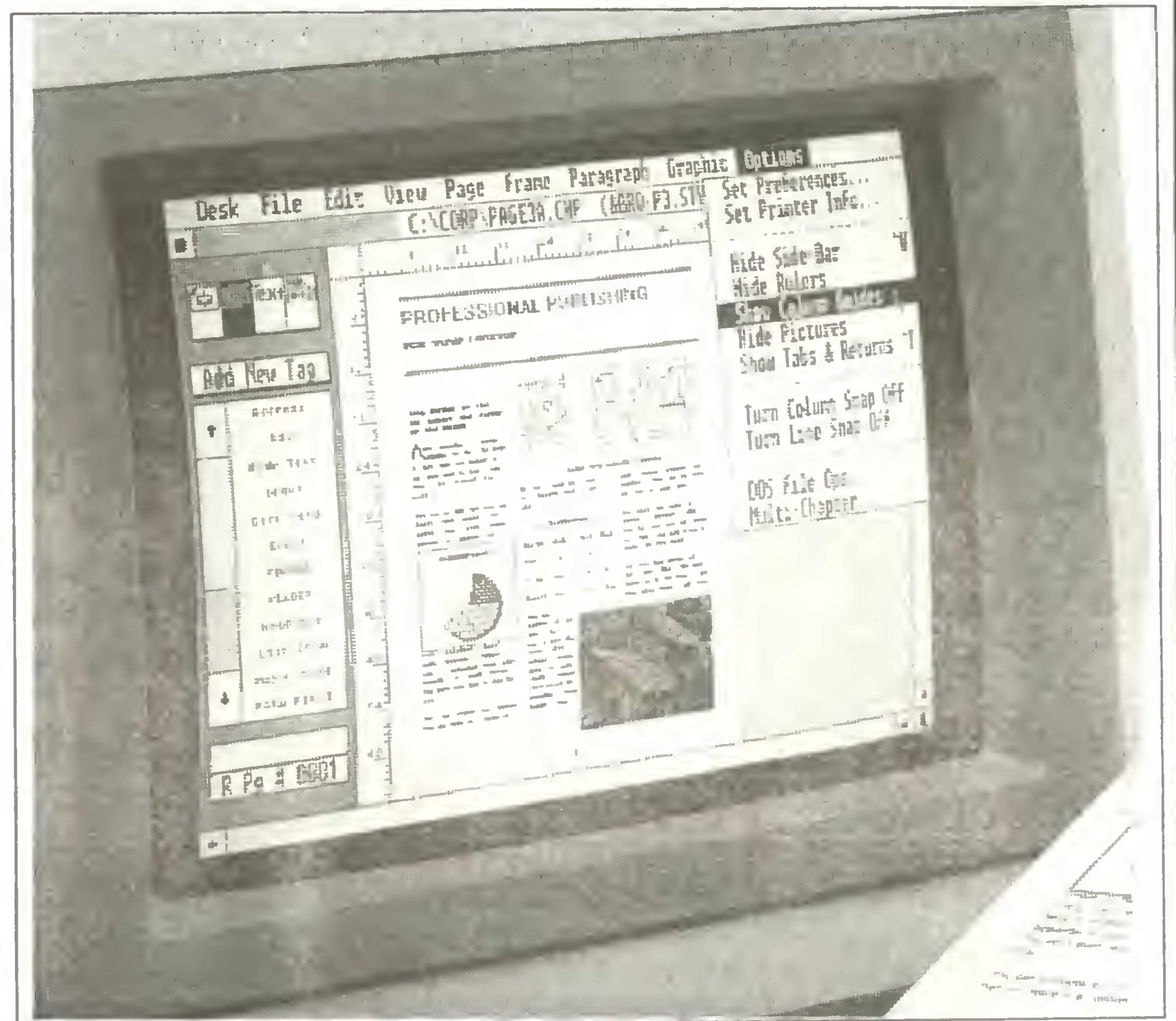
Zapotrzebowanie na "słowo drukowane" jest w społeczeństwie ogromne i to we wszystkich dziedzinach życia: nauce, kulturze, technice, sporcie, komunikacji, rozrywce.

Sytuacja taka spowodowała powstanie oprogramowania pozwalającego stosować komputer do sporządzania i "produkcji" różnych druków. Pierwszymi programami były edytory tekstu. Komputer z programem edytora tekstu zastępował maszynę do pisania. Pozwalał na szybszą i znacznie wyższej jakości pracę, a także na magazynowanie pisanych tekstów i ich ponowne użycie w zmodyfikowanej postaci bez konieczności ciągłego przepisywania.

Dalszy rozwój oprogramowania to tworzenie warunków umożliwiających pełną obróbkę typograficzną tekstów. Kolejny etap to łączenie tekstów i grafiki na stronie. Tak powstały programy typu *Desktop Publishing* popularnie zwane drukarnią na biurku.

Co oferuje typowy program *Desktop Publishing*? Umożliwia typograficzną obróbkę tekstu napisanego za pomocą dowolnego edytora. W przypadku przejmowania tekstu nie ma znaczenia typ komputera na jakim tekst został napisany. Problemy mogą wystąpić jedynie w transmisji zbiorów tekstowych między różnego typu nośnikami pamięci zewnętrznych. Program umożliwia też połączenie tekstu z grafiką tworzoną za pomocą programów graficznych dostępnych najczęściej tylko dla typu komputera, na którym pracuje nasz program DTP. Obsługa typograficzna tekstu polega na uformowaniu go w sposób określony przez użytkownika programu DTP. Użytkownik może wybierać krój i stopień pisma, określić sposób łamania tekstu, rozmieszczenie tytułów oraz ich wielkości, położenie ilustracji i ich wymiary. Można także korzystać z funkcji dodatkowych, jak automatyczne numerowanie stron, wypisywanie pagin, rysowanie linii dzielących szpalty czy funkcji pozwalających na przeliczanie obrabianego tekstu. Operator programu DTP wybiera też jednostki miar typograficznych, w jakich podaje parametry składu i ustala wymiary wszystkich elementów publikacji.

Programy typu DTP skonstruowane są tak, że pozwalają na ciągłą kontrolę realizowanych funkcji. Na ekranie monitora komputera widoczna jest zawsze wybrana strona obrabianej publikacji z zastosowaniem funkcji WYSIWYG (widać tak jak będzie drukowane). Taki sposób prezentacji ułatwia pracę i częściowo zwalnia użytkownika od dokładnego wcześniejszego przeliczania tekstu. Prace nad kompozycją strony kończą się wydrukiem publikacji. Jakość wydruku zależna jest od typu urządzenia wyjściowego – drukarki. Jeżeli drukarka jest urządzeniem popularnym (drukarka igłowa) to kopia nie będzie estetyczna i będzie się różniła od "produkcji drukarnianej". Jeżeli zaś użyjemy drukarki laserowej, uzyskamy kopie dobrej jakości, które mogą być reprodukowane i powielane w wielu egzemplarzach (np. biuletyny, instrukcje obsługi, gazetki okolicznościowe itp.).



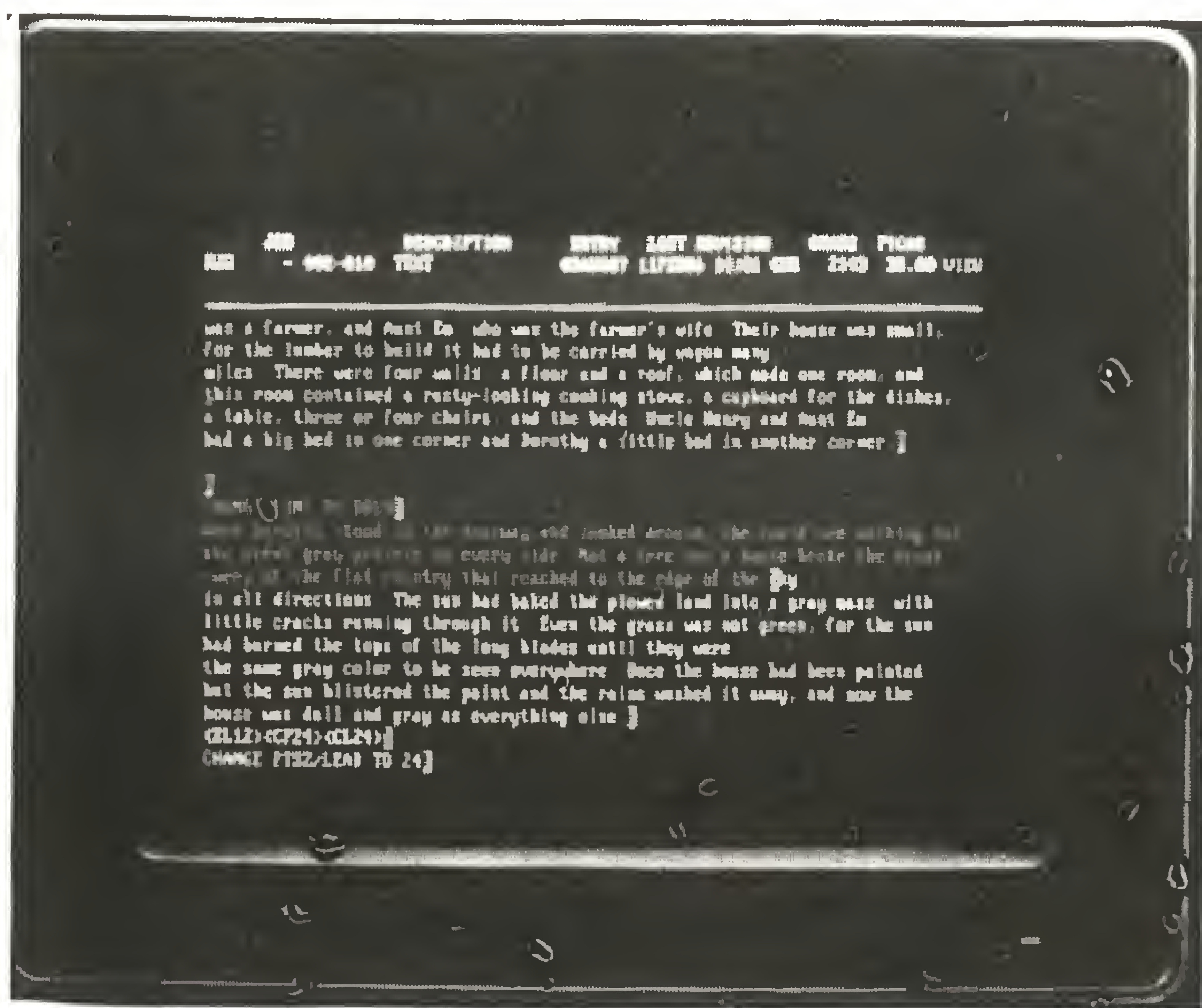
Przykład okienkowych, wybieranych myszką rozkazów obsługi typowego programu DTP dla komputerów standardu IBM PC (Ventura firmy Xerox).

Najtrudniejszym etapem pracy z programami DTP jest zmieszczenie tekstu w zadanym miejscu publikacji. Najczęściej jest go za dużo lub za mało. W przeszłości (uwaga ta dotyczy tzw. gorącego składu) gdy brakowało kilku wierszy tekstu do wypełnienia miejsca, które dla niego przewidziano, zadaniem redakcji czy twórcy publikacji było dopisanie nowego fragmentu. W przypadku gdy tekst był za długi, należało go skracać. Osoba zajmująca się tekstami musiała dobrze znać miary drukarskie i parametry używanych w tworzonej publikacji krojów pism.

W programach DTP sytuacja wygląda inaczej. Gdy tekst jest za długi, operator musi dokonać analizy i zmienić parametry składu np. wielkość interlinii (odstępów między wierszami) lub wielkość liter. Po podaniu nowych parametrów składu komputer na nowo przetwarza tekst i umieszcza go we wskazanym miejscu. Jeżeli tekst jeszcze się nie mieści, dobór parametrów składu należy powtórzyć.

Zastosowanie komputera pozwoliło na płynne lub z bardzo małym skokiem operowanie wielkością odstępów między literami i wierszami składu, co było możliwe w bardzo małym zakresie przy składaniu tekstów z odlewanych pojedynczych czcionek czy ca-

> 38



Przykład typowej tekstowej obsługi nowoczesnego fotoskładu. Na zdjęciu najnowsze rozwiązanie firmy Berthold wykorzystujące komputery IBM PS2/50 jako terminale fotonaświetlarki

37

łych wierszy linotypowych. Przy komputerowej obróbce tekstu można pozwolić sobie na kilkakrotne próby z różnymi parametrami składu, czynność ta "nic nie kosztuje" i nie pociąga za sobą kolejnego przepisania tekstu, jak miało to miejsce przy ręcznej pracy zecera czy operatora linotypu. W przypadku składu gorącego każda zmiana parametrów pociąga za sobą odlanie nowych wierszy tekstu i ponowne ułożenie ich w szpalty i kolumny. Stąd konieczność bardzo dokładnego obliczania wielkości tekstu i miejsca na niego przeznaczonego.

Oprócz wielu zalet i możliwości, programy DTP mają ograniczenia powodujące, że tworzone publikacje nie zawsze przypominają formy drukowane i przygotowywane przez profesjonalne stanowiska fotoskładowe. DTP ma małą liczbę dostępnych jednocześnie krojów pism i ich wielkości. Wynika to z ograniczeń pojemności pamięci operacyjnej komputerów lub z bardzo pracochłonnego procesu przygotowania nowego kroju pisma. Ma to wpływ szczególnie na wyróżnienia tekstu, takie jak pogrubienie, pisanie czcionką pochyłą lub literami zwężonymi czy poszerzonymi. Aby uzyskać takie wyróżnienia, programy DTP drukują wybrane litery danego kroju pisma przez powielanie lub minimalne przesunięcie punktów, z jakich zbudowany jest ich wizerunek. Pismo takie nie ma nic wspólnego z zasadami graficznymi tworzenia krojów pism, a rozwiązanie to jest tylko koniecznym kompromisem.

Innym ograniczeniem jest konieczność sprawdzania efektów pracy programu po zadaniu nowych parametrów składu. Każda zmiana musi być przez operatora sprawdzona i akceptowana.

Profesjonalne stanowiska fotoskładowe to obecnie bardzo szybkie wieloprocesorowe komputery specjalizowane. Wykonują w zasadzie te same czynności co zaawansowane programy typu DTP (programy DTP swym rodowodem sięgają do założeń oprogramowania maszyn fotoskładających). Mają jednak znacznie większe możliwości. Po pierwsze, przy zastosowaniu jako pamięci zewnętrznej szybkich dysków twardych lub napędów dla taśm magnetycznych, prawie nieograniczony staje się zbiór dostępnych na każde życzenie krojów i wielkości pisma. Po drugie, zastosowanie kilku procesorów w jednej maszynie pozwala na zadanie parametrów składu i podanie miejsca przeznaczonego na tekst, a komputer sam zmodyfikuje wcześniej ustalone parametry tak, aby tekst w całości wypełnił przypisane mu miejsce. Ponadto nowoczesna maszyna składająca posługuje się literami opisanymi wektorowo, a nie mapą bitową, jak ma to miejsce najczęściej w programach DTP. Opis wektorowy pozwala na bezstopniowe powiększanie lub zmniejszanie liter. Umożliwia też zmianę szerokości liter danego kroju. W jednym wierszu więc można umieścić większą lub mniejszą liczbę liter, a co za tym idzie zmieszczenie całego tekstu w wyznaczonym miejscu bez zmiany interlinii czy wysokości liter. Postępowanie takie praktycznie nie jest możliwe w większości programów DTP. Kolejną przewagą drukarnianych stanowisk składających jest dostęp do znacznego zestawu znaków specjalnych. Liczba tych znaków zależna jest od pojemności pamięci zewnętrznej.

W ostatnim czasie zawrotną karierę w zastosowaniach DTP robi język opisu wektorowego zwany PostScript opracowany przez firmę Adobe. System ten pozwala na całkowicie wektorowy opis liter, używanych linii, ramek a także ilustracji. Stosując ten system programy DTP swymi możliwościami przypominają profesjonalne stanowiska fotoskładowe. Użycie języka PostScript pozwala także na sterowanie drukarnianą, bardzo wysokiej rozdzielczości (do 1200 linii na cm), naświetlarką. Możliwe jest pełne opracowanie publikacji "w domu" i przyniesienie jej na dyskietce do drukarni. Wykonanie wszystkich poleceń programu w języku PostScript przez jego drukarniany interfejs spowoduje uzyskanie pełnowartościowych klisz stanowiących podstawę do wykonania form drukowych.

Stosowanie tego systemu może w niedługim czasie zrewolucjonizować pracę drukarń, a przede wszystkim pracę redakcji wielu czasopism. Kilka komputerów w redakcji (mogą to być komputery standardu IBM PC, Apple Macintosh, Atari Mega ST) wraz z oprogramowaniem typu DTP pracującym w języku PostScript pozwala na całkowite ominięcie składu w drukarni oraz na bardzo znaczne przyśpieszenie i aktualizację wydawanego pisma. Zadaniem drukarni byłoby, w tym przypadku, wykonanie naświetleń klisz, wykonanie form drukowych i druk czasopisma. Aby ten sposób pracy mógł być stosowany, drukarnie muszą być wyposażone w procesory – interfejsy języka PostScript pośredniczące między stacją przyjmowania danych (dyskietki lub łącza telefoniczne) i naświetlarką laserową. Urządzenia takie są już szeroko oferowane i nie są drogie. W wielu drukarniach na świecie stanowią niezbędne wyposażenie każdego fotoskładu. W naszym kraju nie ma jak dotąd żadnego takiego urządzenia.

Tą smutną uwagą kończymy cykl mający na celu przybliżenie Czytelnikom kilku zagadnień związanych, jak mawia nasz redakcyjny grafik "z kulturą zadrukowanego papieru". Ten zadrukowany papier to element cywilizacji i niezbędny atrybut kultury ogólnoludzkiej. Aby kultura ta stała na jak najwyższym poziomie, trzeba uświadomić sobie jej korzenie i ciągły rozwój. Mamy nadzieję, że treść cyklu pozwoliła wszystkim, którzy mają dostęp do nowoczesnej techniki komputerowej, a przez nią do "papierowej komunikacji", na zrozumienie i stosowanie wypracowanych przez wieki zasad.

W pracy

Zenon Rudak

Jeszcze mniejszy PC

W ubiegłym roku pisałem o, jak się wtedy wydawało, najmniejszym komputerze standardu IBM PC. Był to komputer firmy Husky. Obecnie chcę przedstawić najnowszy produkt znanej na naszym rynku firmy Atari. Nazywa się *Atari folio* i nie byłoby w nim nic ciekawego, gdyby nie jego wymiary oraz możliwości. Komputer ten ma postać składanego pudełka przypominającego zeszyt lub notatnik o wymiarach: długość 20 cm, szerokość 10 cm i wysokość 3 cm.

Nowy produkt Atari skonstruowany został zgodnie z panującą w krajach wysoko rozwiniętych "modą" na przenośne lub wręcz kieszonkowe komputery, zezwalające na pracę w każdych warunkach. Urządzenia takie mają zazwyczaj mały ciekłokrystaliczny ekran, podtrzymywaną bateryjnie pamięć RAM oraz szereg programów użytkowych zapisanych w pamięci ROM. Większość takich konstrukcji pozwala na podłączenie do sieci telefonicznej i przesyłanie bądź przyjmowanie danych z innych komputerów lub ogólnodostępnych banków informacji.

Atari folio wyposażony jest w procesor Intel 80C88. Procesor pracuje z zegarem o częstotliwości 4,91 MHz. Zegar nie może pracować z większą częstotliwością, przez co maszyna nie należy do komputerów szybkich. Procesor współpracuje ze 128 KB pamięci RAM. Obszar ten może być podzielony przez użytkownika na pamięć operacyjną i RAM-dysk. Podział jest dowolny. Procesor korzysta z pamięci ROM o pojemności 256 KB. Zawarte są tu procedury BIOS typowe dla komputerów standardu IBM PC, procedury dodatkowe służące do obsługi ekranu, klawiatury, systemu zapisu i przechowywania informacji. W pamięci ROM *Atari folio* zapisane jest także bogate oprogramowanie użytkowe.



Folio ma mały ekran ciekłokrystaliczny. Ekran pracuje tak jak karta MDA (monochromatyczna karta graficzna) typowego komputera IBM PC. Pozwala wyświetlać 40 znaków tekstu w 8 wierszach lub grafikę z rozdzielczością 240 na 64 punkty. Procedury obsługi ekranu pozwalają na programowe przełączenie ekranu na wyświetlanie 80 znaków tekstu w 8 wierszach. Pozwalają one także na oglądanie całego obrazu karty MDA. Ekran Atari wyświetla jednorazowo wycinek stanowiący ok. jednej szóstej całego obrazu. Aby obejrzeć cały obraz, wykorzystuje się możliwości "przesuwania" okna ekranu po obrazie. Ekran *Atari folio* ma możliwość zmiany kontrastu znaki – tło. Zmian tych dokonuje się programowo rozkazem z klawiatury. Dostępnych jest osiem poziomów ustawienia kontrastu świecenia ekranu.

Komputer wyposażony jest w klawiaturę podobną do przycisków stosowanych w popularnych kalkulatorach. Zawiera ona wszystkie znaki liter i cyfr, klawisze sterujące ruchem kursora, Esc, Ctrl, Alt stosowanych w standardzie IBM PC. Klawiatura nie ma zestawu klawiszy funkcyjnych oraz pola klawiszy numerycznych (zwykle w standardowych klawiaturach umieszczonego po prawej stronie pola liter) służącego do szybkiego wprowadzania danych liczbowych. Funkcje klawiszy funkcyjnych wywołuje się poprzez równoczesne naciskanie klawisza specjalnego i klawiszy z cyfrą odpowiadającą numerowi klawisza funkcyjnego. Klawisz specjalny oznaczony jest na klawiaturze znakiem firmowym Atari. Pole klawiszy numerycznych zostało umieszczone na klawiaturze liter. Prawa część tej klawiatury została dodatkowo oznakowana cyframi (trzecia funkcja klawisza). Działanie tej mini-klawiatury numerycznej możliwe jest dopiero po naciśnięciu kombinacji trzech klawiszy, podobnie jak w komputerze ZX Spectrum 48K. Różnica polega na zastosowaniu klawiszy z twardego tworzywa sztucznego.

Atari folio nie ma napędu dyskowego ani sterownika pozwalającego na korzystanie z niego. Zapisywanie danych i ich przechowywanie możliwe jest w wydzielonym z pamięci operacyjnej RAM-dysku lub w zewnętrznej pamięci ROM. (System zapisywania informacji jest bardzo podobny do opisywanego w teście komputera Psion Organiser II). W lewej bocznej ściance obudowy *Atari folio* znajduje się gniazdo do specjalnych kart pamięci zewnętrznej. Karty te wyposażone są w układy EPROM o pojemności 32 lub 128 KB. Pamięci te pozwalają na jednokrotny zapis informacji i wielokrotny jej odczyt. Aby umożliwić ponowny zapis lub zmianę danych karty, należy skasować, tak jak robi się to przy czyszczeniu układów EPROM.

Komputer ma jedno złącze umożliwiające dołączenie modułów rozszerzenia. Podstawowymi modułami rozbudowy maszyny są: moduł pamięci RAM pozwalający otrzymać łącznie obszar 640 KB, moduł interfejsu szeregowego i równoległego zapewniającego korzystanie z drukarki i modemu telefonicznego. Dalsza rozbudowa *Atari folio* polega na podłączeniu modułu transmisji danych, do lub z komputerów standardu IBM PC. Drugim modułem transmisyjnym jest karta przeznaczona do instalacji w komputerze PC/XT/AT, umożliwiająca odczytywanie lub zapisywanie kart pamięci zewnętrznej *Atari folio*. Interfejs ten pozwala na łatwe i szybkie przenoszenie danych lub programów między nowym produktem Atari a komputerami stacjonarnymi. Komputer nie ma wyjścia dla monitora zewnętrznego, nie przewidziano także możliwości podłączenia innej klawiatury.

Atari folio zaopatrzony jest w bogate oprogramowanie użytkowe. Wszystkie programy zapisane są w pamięci ROM i dostępne w każdej chwili. W pamięci ROM zapisany jest system operacyjny DOS, zastosowano wersję 2.11, która była opracowana przez firmę Microsoft dla komputerów typu *laptop* (przenośne, zasilane bateryjnie komputery walizkowe). Poza systemem operacyjnym w pamięci ROM znajdują się programy: zarządzania modułami rozszerzeń, elektronicznego arkusza obliczeniowego, elektronicznej sekretarki przypominającej ważne informacje, elektronicznej książki telefonicznej połączonej z bazą adresów, rozbudowanego kalendarza stuletniego umożliwiającego przekazywanie informacji w ustalonym wcześniej czasie, nieskomplikowanego edytora tekstu i programu do obsługi transmisji danych między *folio* a komputerami stacjonarnymi.

Na uwagę zasługuje program elektronicznego arkusza obliczeniowego. Pozwala on na pracę z plikami tworzonymi przez znany pakiet programowy Lotus 1-2-3 w wersji 1.0 i 2.01. Drugą ciekawostką w programach wbudowanych w pamięć ROM *Atari folio* jest generator dźwięku połączony z elektroniczną książką telefoniczną. Kombinacja ta pozwala na odszukiwanie numerów telefonów ze spisu ułożonego przez użytkownika i podawanie dźwięku wywołującego dany numer do słuchawki telefonu. Pozwala to szybko uzyskać połączenie z wieloma abonentami bez konieczności ciągłego wykręcania numerów (oczywiście system ten działa, gdy sieć telefoniczna jest do tego przystosowana). Głośnik emitujący dźwięk zamontowany jest w otwieranej pokrywie z lewej strony ciekłokrystalicznego ekranu. Program edytora tekstu ma funkcje szukaj – zamień, liczenia znaków w tekście oraz druku tekstu przy podłączeniu odpowiedniego modułu rozszerzenia.

Komputer zasilany jest z baterii typu R6 lub zasilacza sieciowego. Włączenie zasilacza odłącza baterie. Odłączenie zasilacza sieciowego powoduje automatyczne uruchomienie systemu oszczędzania baterii, który wygasza ekran komputera, gdy przez minutę nie są wykonywane żadne operacje ekranowe. Przyciśnięcie dowolnego klawisza lub wprowadzenie nowych danych powoduje przywrócenie poprzedniej zawartości ekranu. Działanie systemu oszczędzania umożliwia pracę przez ok. 160 godzin na jednym komplecie baterii.

Komputer *Atari folio* jest niewątpliwie ciekawym urządzeniem, przeznaczonym dla menedżerów, handlowców, dziennikarzy. Pozwala pracować w czasie podróży, narad, bezpośrednich kontaktów z klientem itp. *Atari folio* ze względu na swoje ograniczenia konstrukcyjne, dotyczące głównie ekranu i małego obszaru pamięci operacyjnej RAM, nie jest w pełni komputerem standardu IBM PC. Jest jego namiastką pozwalającą jednak na prowadzenie prostych prac z tekstami czy małymi zbiorami danych. Uciążliwa dla użytkownika wydaje się konieczność korzystania z szeregu dodatkowych interfejsów dla zapewnienia normalnej pracy maszyny (korzystanie z trwałego zapisu danych, drukowania zbiorów, podłączenie do sieci komputerowych). W takim przypadku korzystniejszy byłby chyba zakup komputera typu *laptop* będącego miniaturową pełną wersją komputera stacjonarnego. Przyszłość pokaże czy tak uproszczone sprzętowo a zarazem o skomplikowanej obsłudze komputery zdobędą rynek.



Dzięki uprzejmości pana Włodzimierza Bielskiego, właściciela londyńskiej firmy wysyłkowej Electronic Export, redakcja mogła testować i przez pewien czas korzystać ze sterownika i monitora całostronicowego firmy Packard Bell. Dziękujemy!

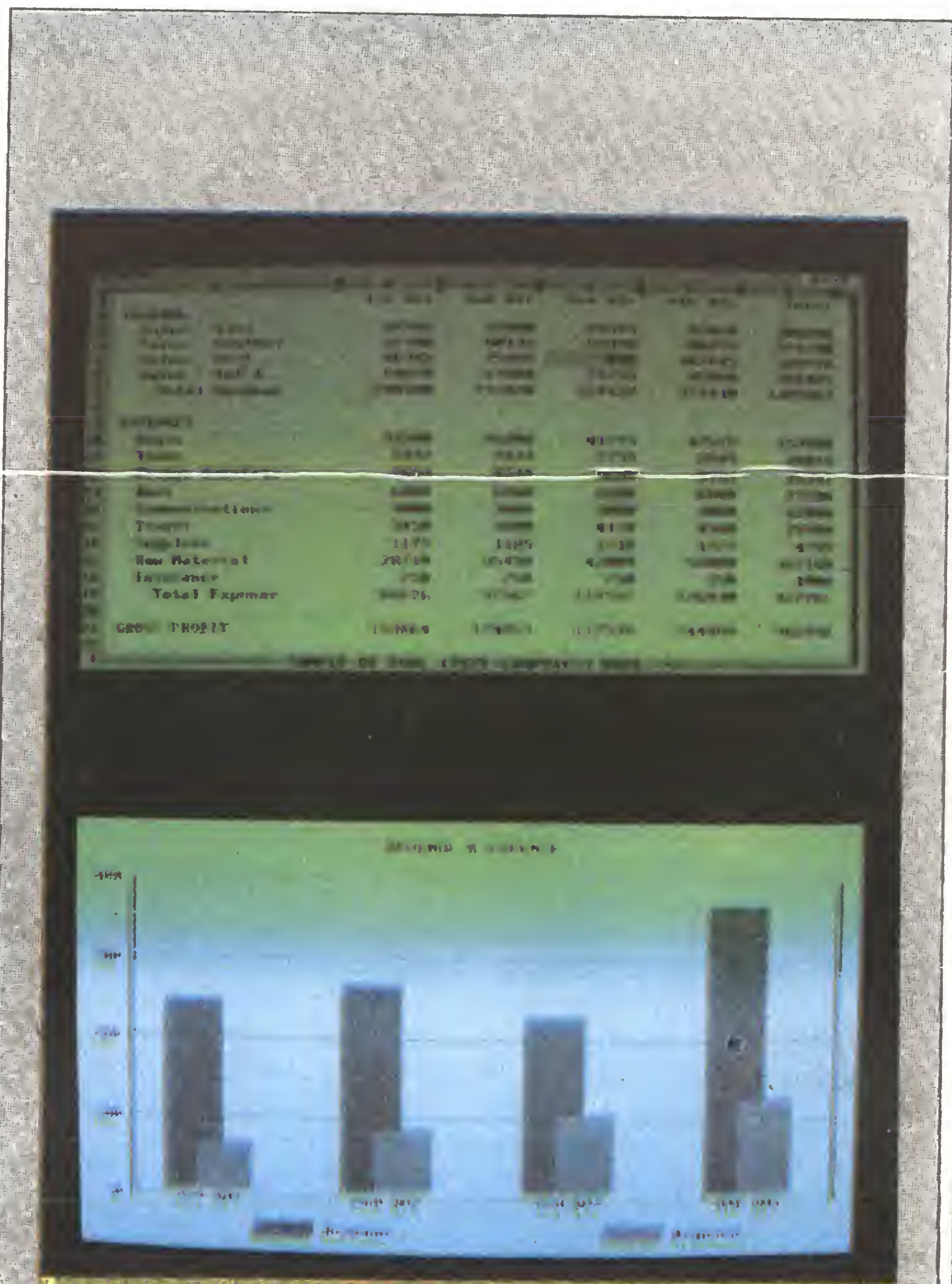
Na początku masowej komputeryzacji stanowisk pracy zwracano głównie uwagę na ilość i dostępność podstawowego sprzętu informatycznego. Najważniejsze było posiadanie komputera, bez względu na to jaka była jego konfiguracja. Mowa tu oczywiście o sprzęcie standardu IBM PC, który dzięki zabiegom koncernu IBM stał się podstawą komputeryzacji. W miarę nasycania rynku sprzętem podstawowym, zaczęto wzbogacać konfigurację komputerów o dodatkowe urządzenia peryferyjne, które instalowano tak, aby stwarzać pełnosprawne samodzielne stanowiska pracy. Spowodowało to szybki rozwój konstrukcji urządzeń peryferyjnych o funkcjach odpowiadających wymaganiom przyszłych użytkowników. W przypadku baz danych były to głównie dodatkowe terminale służące do wprowadzania danych i nośniki pamięci zewnętrznych o dużej pojemności i dużej szybkości działania. W stanowiskach do prac projektowych niezbędne stawały się *plotery* i *digitizery*. Do prac nad przetwarzaniem tekstów konfigurowano komputer tak, aby operator miał przed sobą wyraźny i czytelny ekran. Oczywiście dobór konfiguracji komputera zależny jest od dostępności dodatkowych urządzeń peryferyjnych. Oferta jest stale wzbogacana o coraz nowsze i doskonalsze konstrukcje. Tutaj przedstawię jedno z takich dodatkowych rozwiązań. Jest nim specjalizowany sterownik obrazu i monitor produkowany przez angielską firmę Packard Bell.

Jak wielu użytkownikom komputerów standardu IBM PC wiadomo, praca nad dużymi zbiorami tekstowymi wymaga ciągłego przesuwania fragmentów tworzonych dokumentów na ekranie monitora. Typowe monitory są w stanie wyświetlać maksymalnie 24 wiersze tekstu. Zawartość ekranu nigdy zatem nie odpowiada standardowej stronie maszynopisu (30 wierszy). Stan ten pogarsza się, gdy musimy tworzyć tabelki lub zadbać o elastyczny wygląd drukowanego dokumentu. Podobnie jest przy pracy z programami obsługującymi bazy danych. Mała liczba wyświetlanych wierszy tekstu, tabel, zestawień, pól arkusza obliczeniowego nie ułatwia pracy i zmusza nieraz do długotrwałego poszukiwania wybranego miejsca czy fragmentu tekstu. Odpowiednim rozwiązaniem byłoby więc użycie takiego monitora, który pozwalałby na wyświetlanie znacznie większej liczby wierszy tekstu, dłuższych tabel, większej ilości pól bazy danych. Takie właśnie rozwiązanie proponuje firma Packard Bell.

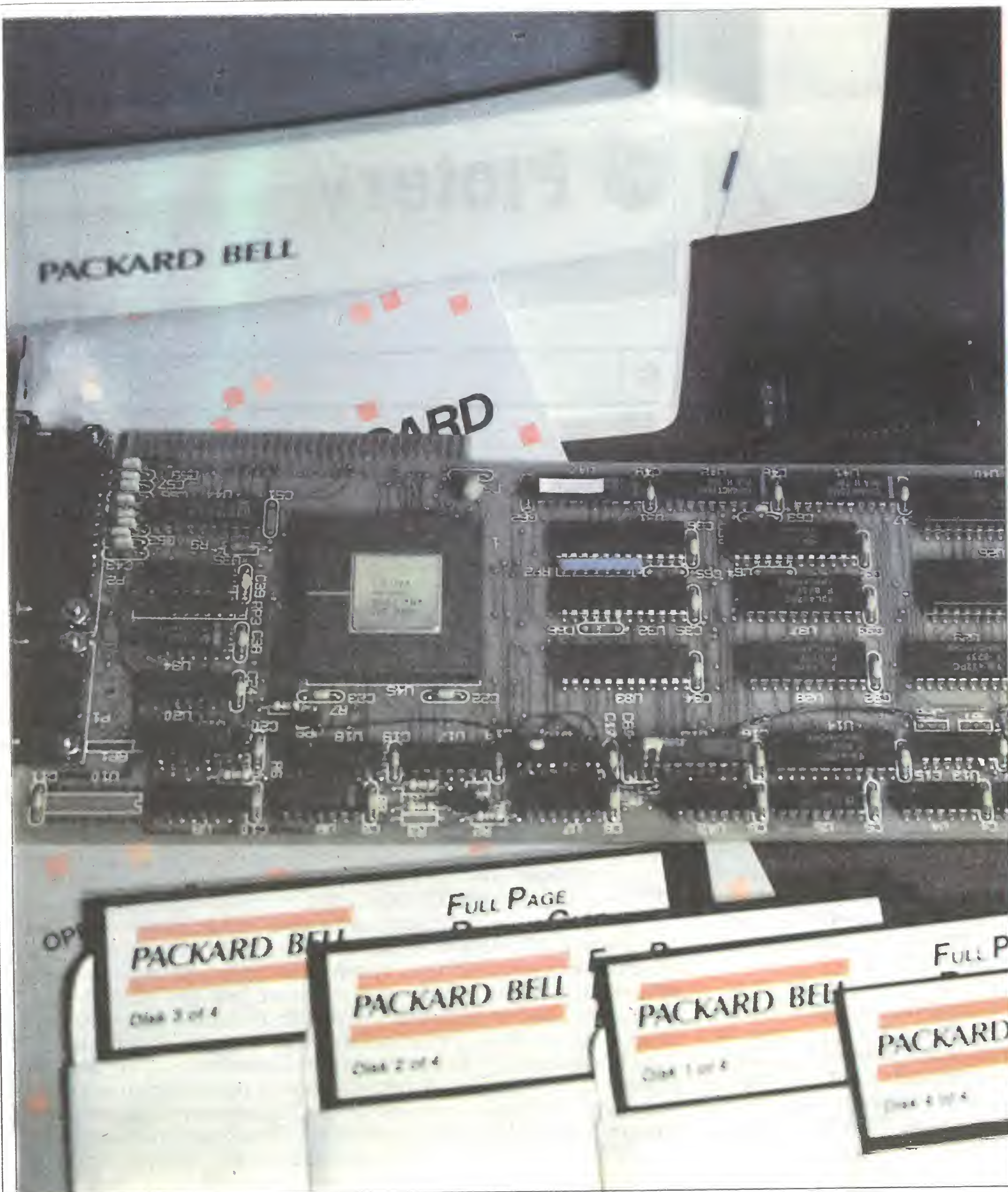
Skonstruowany w firmie sterownik obrazu pozwala na wyświetlanie do 66 wierszy tekstu, pozwala także na pracę graficzną ekranu z rozdzielczością 738 na 1008 punktów. Obraz jest monochromatyczny i może być wyświetlany w 32 - stopniowej skali szarości. Ze sterownikiem współpracuje monitor o powierzchni ekranu równej arkuszowi formatu A4 w układzie pionowym. Świecące punkty ekranu monitora mają kolor biały. Powierzchnię ekranu wykonano z matowego szkła. Sterownik obrazu umieszczony jest na pełnowymiarowej karcie rozszerzenia standardu IBM PC. Jest on specjalizowanym komputerem i ma za zadanie obsługę współpracującego z nim monitora.

Sterownik zawiera procesor LIA 3728. Procesor pracuje z zegarem o częstotliwości 8 MHz. Ze względu na znacznie większą liczbę wyświetlanych znaków i punktów graficznego obrazu procesor posługuje się pamięcią RAM o pojemności 512 KB oraz dodatkowo 16 KB pamięci statycznej RAM. Pamięć RAM jest pamięcią obrazu oraz pamięcią operacyjną procesora. Sterownik pracuje według programu (BIOS sterownika) zapisanego w pamięci ROM. Do komunikacji sterownika z procesorem głównym komputera wykorzystywana jest magistrala systemowa oraz procedury z BIOS komputera. Sterownik dodatkowo wyposażony jest w interfejs równoległy typu Centronics do obsługi drukarki. Interfejs ten po zainstalowaniu karty w komputerze jest zawsze "widoczny" dla komputera jako pierwszy port równoległy (LPT 1).

Zastosowanie specjalizowanego procesora obrazu pozwoliło na uzyskanie dodatkowych możliwości prezentacji wyświetlanego tekstu. Wykorzystywany jest do tego celu bogaty zestaw atrybutów znaków prezentowanych na ekranie. Określenie atrybutów znaków umożliwia realizację funkcji WYSIWYG (w wolnym przekładzie: to co widzisz na ekranie, tak będzie wyglądało na papierze) bardzo pomocnej przy opracowywaniu druków sprawozdań, zestawień, tabel itp. Na podkreślenie zasługuje fakt, że funkcja ta realizowana jest w trybie znakowej pracy sterownika i programu edycji tekstu. Tak więc na ekranie można zobaczyć pogrubione fragmenty tekstu pisane czcionką pochyłą. Można pisać wykładniki i indeksy, można podkreślać lub nadkreślać wybrane znaki, można pisać je w negatywie lub rozjaśnione itp. Sposób wykorzystania atrybutów znaków zależy od możliwości edytora tekstu. Oprócz pracy z tekstem czy bazami danych sterownik Packard Bell z monitorem może być wykorzystany przy programach graficznych. Można wtedy korzystać z rozdzielczości ekranu wynoszącej 738 punktów w poziomie na 1008 punktów w pionie. Oprogramowanie systemowe sterownika pozwala także na pracę w trybie podwójnego ekranu. Górna połowa ekranu monitora wyświetla tekst, tak jak



test COMPUTER



nych zbiorów instalacyjnych na dyskietce lub dysku twardym użytkownika. Aby monitor zaczął poprawną pracę weług ustaleń użytkownika, konieczne jest resetowanie komputera po wykonaniu do końca procedurze instalacyjnej. Po przeprowadzeniu podstawowej instalacji użytkownik może pracować z systemem operacyjnym DOS, wykorzystując możliwość wyświetlania na ekranie 66 wierszy tekstu. Podczas takiej pracy możliwe jest także wykorzystanie atrybutów wyświetlanych znaków (kody stosowane przez program ANSI.COM (sys)). Dostosowanie posiadanego oprogramowania do możliwości monitora polega na wybraniu z dyskietek dołączonych do zestawu karta sterownika – monitor, odpowiednich programów (*driverów*) pośredniczących w obsłudze ekranu. Dla ułatwienia korzystania z programów pośredniczących w instrukcji obsługi sterownika zawarto szczegółowe opisy sposobu instalacji kilkunastu najpopularniejszych programów do współpracy ze sterownikiem Packard Bell. Opisano sposób instalacji (łącznie ze zmianami w kodzie programów) kilku najpopularniejszych edytorów tekstu, programów graficznych, pakietów zintegrowanych, baz danych, nakładek na system operacyjny i programów typu *Desktop Publishing*. Dyskietki z programem instalacyjnym i programami pośredniczącymi zawierają dodatkowo program testujący wszystkie funkcje oferowane przez sterownik i efektowny program demonstrujący jego możliwości.

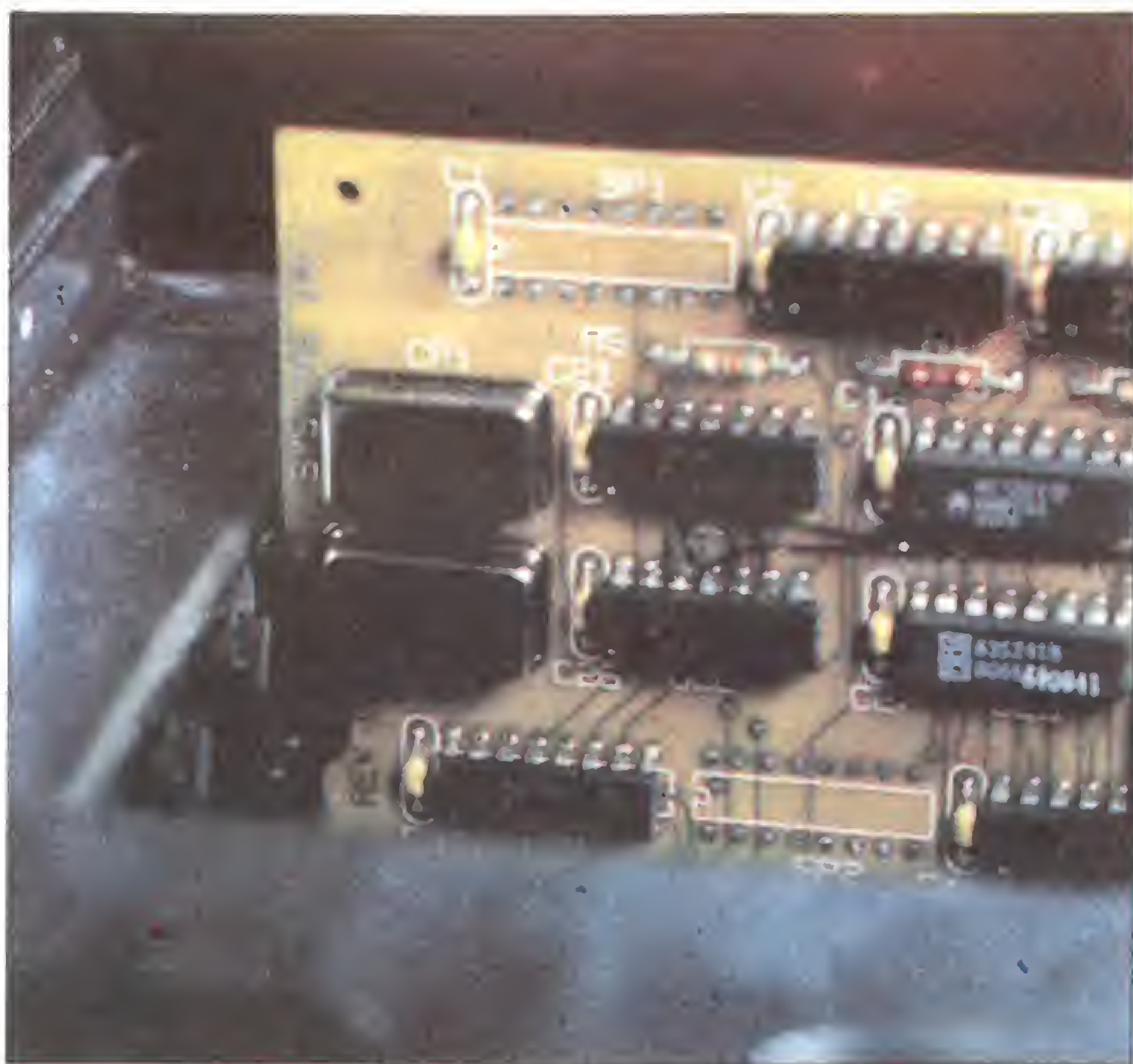
W czasie pracy ze sterownikiem Packard Bell posługiwałem się dostosowanym do jego możliwości oprogramowaniem pracującym w środowisku MS Windows, GEM i MS DOS. Pracowałem z edytorami tekstu: PC Write 2.5 i 2.71, WordStar wersje od 2.0 (środowisko MS DOS), MS Write (środowisko MS Windows) i Word Plus (środowisko GEM). Każdy z programów pozwalał na wyświetlanie 66 wierszy tekstu. Aktywne były wszystkie funkcje tych programów, poprawnie przebiegał proces drukowania opracowywanych dokumentów i komunikacja z pamięcią masową (dyskietki, dysk twardy). Praca z programami graficznymi pozwalała na bardzo dokładne opracowywanie rysunków (wysoka rozdzielczość graficzna ekranu). Monitor umożliwiał bardzo dokładne zaplanowanie położenia rysunku na stronie, następnie drukowanej na drukarce. Przy stosowaniu monitora typowego jest to bardzo trudne lub dla niektórych programów niemożliwe. Największe jednak zalety testowanego sterownika i monitora zauważa się przy pracy z programami typu *Desktop Publishing*. Możliwość komponowania strony w jej naturalnej wielkości pozwala uniknąć wielu pomyłek, ułatwia "włamywanie" ilustracji w tekst, dopracowywanie szczegółów graficznych tworzonej publikacji. Dostępne dla

> 42

karta MDA standardu IBM (720 na 348 punktów), a dolna – grafikę z rozdzielczością karty CGA standardu IBM (640 na 200 punktów). Pracą dwóch monitorów może sterować znany w Polsce program wspomagający projektowanie typu AutoCad lub zintegrowany pakiet programowy Lotus 1-2-3. W obu przypadkach górna część ekranu przeznaczona jest na teksty komunikatów lub poleceń, a dolna część wyświetla realizację graficzną wprowadzanych uprzednio danych czy poleceń.

Instalacja sterownika i monitora w dowolnym komputerze jest prosta i polega na zastąpieniu sterownikiem Packard Bell dotychczasowej karty graficznej. Karta Packard Bell wymaga jednego złącza 8-bitowego (standard IBM XT) płyty głównej komputera. Po zamontowaniu karty należy podłączyć do niej monitor. Do połączenia monitora z kartą przeznaczony jest przewód zakończony dwoma wtykami typu szufladowego. Wtyki są różne, co jednoznacznie określa gniazda w monitorze i karcie sterownika. Po podłączeniu monitora i uruchomieniu komputera należy przeprowadzić procedurę programowej instalacji sterownika. Bez instalacji sterownik pracuje tak jak karta Hercules, a obraz wyświetlany jest tylko w górnej połowie ekranu. Program instalacyjny dołączony jest do sterownika wraz z instrukcją na dyskietce. Instalacja przebiega w formie odpowiedzi na zadawane przez komputer pytania. Wybierając kolejne opcje użytkownik decyduje o sposobie wykorzystania sterownika i monitora. Instalacja kończy się zapisaniem trzech zbiorów na dyskietce lub dysku twardym, z którego uruchamiany jest komputer oraz na modyfikacji istniejących zbiorów config.sys i autoexec.bat. Zbiory oryginalne zachowywane są ze zmienioną nazwą, o czym informuje program instalacyjny w czasie kreacji włas-

KOMPUTER test



41 <

mnie programy typu DTP *Desktop Publisher*, *Ventura* (GEM) i *PageMaker* (MS Windows) pracowały poprawnie wykorzystując w pełni możliwości sterownika i monitora. Zainstalowany na karcie sterownika Packard Bell interfejs równoległy typu Centronics dawał poprawne wydruki tekstów i grafiki. Opcja systemu operacyjnego DOS, drukowania znakowej kopii ekranu realizowana jest przez krótki program dołączany w czasie instalacji programowej sterownika do dyskietki uruchamiającej komputer. Program ten wywoływany jest tak samo jak opcja systemu operacyjnego, z tą różnicą, że drukowana jest cała zawartość znakowa rozszerzonego ekranu. Wydruk zajmuje całą kartkę formatu A4.

W czasie instalowania karty sterownika w kilku komputerach, zauważyłem pewną wadę montażową testowanego egzemplarza sterownika. Otóż na krawędzi bocznej tyłu karty umieszczone są rezonatory kwarcowe zegara procesora. Jeden z nich umieszczony jest tak, że zawsze opiera się na prowadnicach montowanych w komputerze, a służących do podtrzymywania zamontowanej karty. Grozi to uszkodzeniem połączeń rezonatora ze ścieżkami drukowanymi karty sterownika. Może mieć to duże znaczenie, gdy karta będzie często montowana w różnych komputerach. Pewnym niedociągnięciem instrukcji obsługi sterownika jest brak dokładnego opisu możliwości wykorzystania przez użytkownika atrybutów znaków.

Na zakończenie należy powiedzieć, że w taki sterownik i monitor powinny być bezwzględnie wyposażane wszelkie stanowiska posługujące się programami *Desktop Publishing*, obsługujące bazy danych i programy graficzne. Nie należy przy tym obawiać się braku oprogramowania umożliwiającego korzystanie z takiego sterownika i monitora. Większość nowych programów dostosowana jest opcjonalnie do wyświetlania obrazu przez całostronicowe monitory. Niezbędne programy lub procedury pośredniczące zawarte są w wersjach instalacyjnych tych programów, a brakujące można znaleźć na dyskietkach dołączonych do sterownika. Stosowanie takich monitorów przy profesjonalnej obróbce tekstów łącznie z przygotowaniem ich do druku przez programy DTP ułatwia pracę, zwiększa wydajność i podnosi jakość tworzonych publikacji.



test COMPUTER



Zbigniew Blewoński

Plotery

Po polsku powinno się mówić autokreślarka, przyjęło się jednak, oczywiście z angielskiego, ploter – czyli zewnętrzne (peryferyjne) urządzenie do kreślenia rysunków "z komputera". Plotery, a właściwie to co było przed nimi, wynaleziono i stosowano chyba nawet wcześniej niż komputery, i to nawet te duże.

Pradziadkiem plotera nazwać można urządzenie rejestrujące graficznie zmienność parametrów elektrycznych w czasie np. przy badaniu EKG. Urządzenie takie jest genialne w swojej prostocie i składa się z mechanizmu przesuwającego (odwijającego) ruchem jednostajnym papier z rolki (oś X) i mechanizmu przenoszącego mierzoną wielkość elektryczną na ruchy pisaka (oś Y). Bezpośrednie przeniesienie tej koncepcji do urządzeń kreślących rysunki techniczne zaowocowało powstaniem w roku 1959 ploterów bębnowych. Następne pojawiły się plotery stołowe. Były one duże, precyzyjne, ale i bardzo drogie – pozwolić sobie mogły na nie tylko duże firmy, podobnie jak i na systemy CAD. Tak było w epoce ICL (ODRA 1304) i IBM 360/370 (RIAD-32).

Rozwój mini- i mikrokomputerów oraz wzrost zainteresowania systemami CAD, które w wydaniu mini- i mikro- stały się dostępne dla całkiem niedużych firm, spowodował znaczny "ruch" w konstrukcjach ploterów. Ruch może oznaczać postęp i obniżkę cen, tak też stało się i w przypadku ploterów. Podobnie jak odkurzacz kojarzy się natychmiast z firmą Elektrolux tak przez długie lata ploter to firma Calcomp. Pierwszy jej model (bębnowy) – Calcomp 565 pojawił się w 1959 roku. W 1965 roku zjawił się Houston Instrument ze swoim małym ploterem stolikowym napędzanym silnikiem krokowym. Hewlett-Packard wkroczył w 1980 roku z rolkowym modelem HP7580, potem przyszły nowe modele tych samych firm oraz wiele innych, nowych firm.

Różnice konstrukcyjne

Podstawową cechą różniącą plotery niskiej jakości od wysokiej jakości jest napęd – silnikiem krokowym lub serwomotorem. W silniku krokowym ruch (obrót) odbywa się skokowo między z góry określonymi położeniami dopuszczalnymi. Silnik taki jest tani, ale przy tym głośny, mało dokładny i wolny. Serwomotor to cały skomplikowany układ silnik – sterowanie elektroniczne i ruch ma płynny. Serwomotor jest drogi, ale za to cichy, dokładny i szybki. Obecnie ceny serwomotorów spadają, rośnie jednocześnie jakość (dokładność) silników krokowych.

Drugą cechą konstrukcyjną różniącą plotery jest sposób poruszania się względem siebie papieru i pisaków. Tradycyjnym rozwiązaniem jest tzw. **ploter stołowy**, w którym papier mocowany jest płasko do stołu i w trakcie rysowania pozostaje cały czas nieruchomy. Mocowanie polega na przyklejeniu przylepcem, przyciśnięciu listwami magnetycznymi lub odbywa się elektrostatycznie czy też pneumatycznie. Wzdłuż papieru przesuwa się (jeden silnik) ramię, po którym przesuwany jest (drugi silnik) pisak. Zaletą tego typu ploterów jest łatwość mocowania papieru oraz możliwość rysowania do samej jego krawędzi, wadą natomiast duża powierzchnia, jaką zajmują.

Innym rozwiązaniem są tzw. **plotery rolkowe** zwane popularnie "wyżymaczkami". Papier mocowany jest w nich przez docisk gumowymi kółkami do wałka o chropowatej powierzchni. W trakcie kreślenia pisak przesuwa się w poprzek papieru (jeden silnik) – oś Y, a papier przesuwany jest w poprzek plotera (drugi silnik) – oś X.

Plotery tego typu zajmują mało miejsca i dominują obecnie w grupie ploterów dużych formatów. Pierwowzorem tego typu były plotery bębnowe, w których specjalny papier (najczęściej perforowany podobnie jak w drukarkach) odwijany był z rolki w trakcie rysowania. Następcami były tzw. **plotery pasowe**, w których arkusz papieru przylepiany był do poruszającego się w trakcie rysowania specjalnego pasa z folii – podobnie jak w przenośniku taśmowym.

Format rysunku

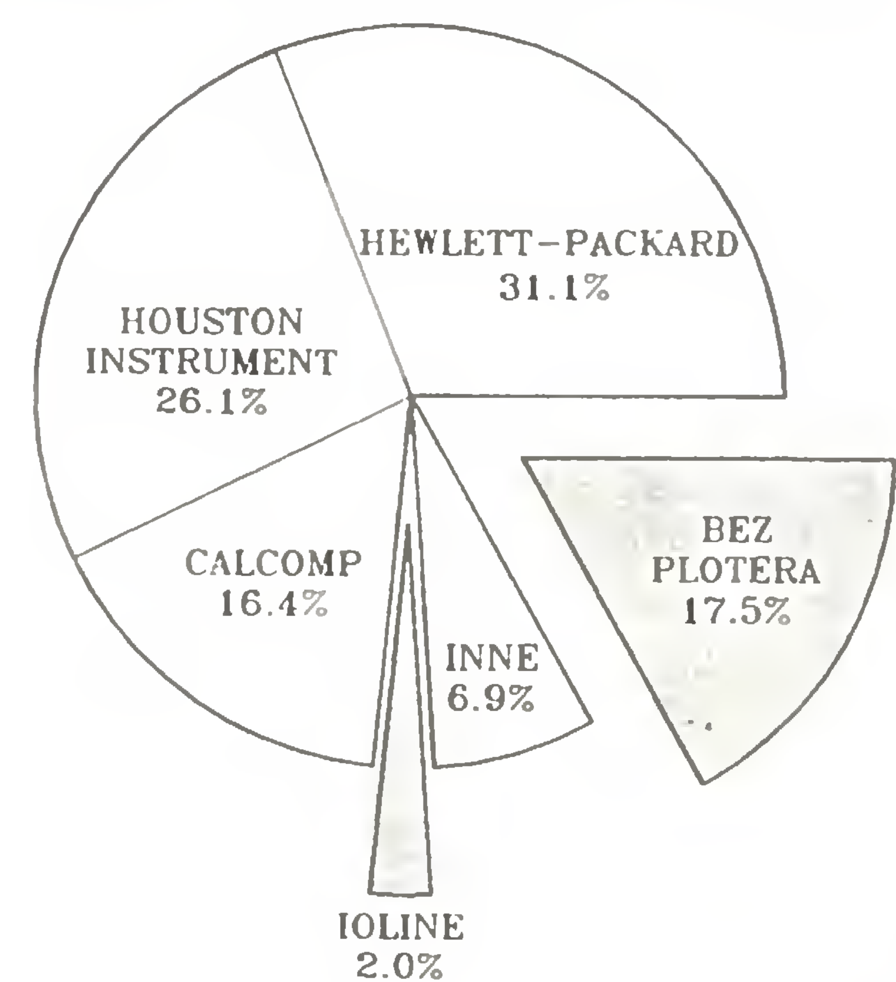
Pierwszym parametrem określającym przydatność plotera do określonych potrzeb jest maksymalny format rysunku, jaki można na nim wykreślić. Format rysunków określany jest zgodnie z normami ANSI (amerykańskimi) lub obowiązującymi i u nas normami ISO/DIN. Wielkości arkuszy według obydwu standardów różnią się nieznacznie (patrz tabelka). Najwięcej firm oferuje plotery formatu A3/A4 – stosunkowo proste do wykonania i tanie. Plotery dużych formatów A0/A1 oferuje znacznie mniej firm, gdyż wymagają znacznie bardziej precyzyjnej "mechaniki", są także znacznie droższe. Małe plotery wykorzystywane są głównie do rysowania wykresów (tzw. *business graphic*) np. z Lotusa 1-2-3, duże do kreślenia rysunków technicznych z systemów CAD. Format wykonywanego rysunku określany jest z programu rysującego (np. AutoCAD), niektóre plotery dysponują czujnikami automatycznie określającymi format rysunku zależnie od wielkości założonego papieru. Plotery dużych formatów pozwalają na kreślenie rysunków we wszystkich formatach mniejszych od maksymalnego. W przypadku ploterów rolkowych papier podawany może być w arkuszach lub automatycznie brany z rolki.

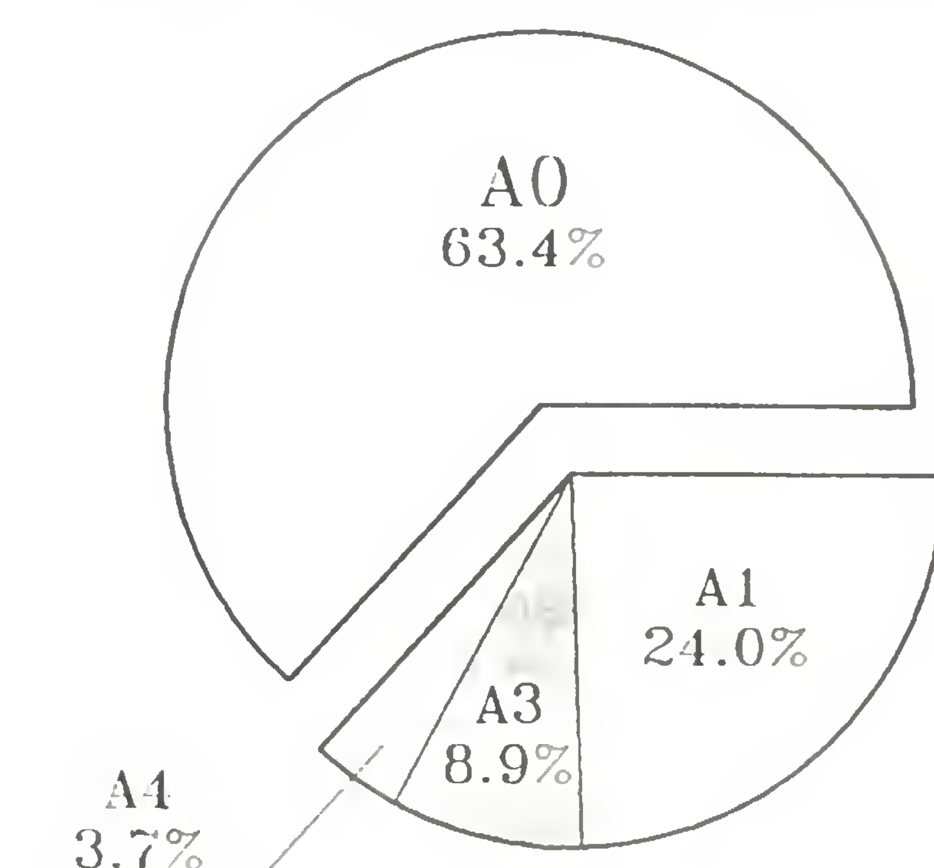
Prędkość ruchu pióra

Maksymalna prędkość ruchu pióra jest miarą szybkości poruszania się pisaka – w niektórych ploterach możliwe jest niezależne sterowanie szybkością kreślenia i szybkością ruchu z uniesionym pisakiem. Dla każdego plotera podana powinna być zawsze tzw. prędkość osiowa czyli prędkość ruchu pisaka wzdłuż jednej osi i tzw. prędkość wypadkowa czyli prędkość jednoczesnego ruchu wzdłuż obu osi. Prędkość podawana jest w calach na sekundę lub w milimetrach na sekundę i zawiera się zwykle w granicach 30-800 mm/s (132 ips). Pamiętać należy, że podawane w reklamach maksymalne prędkości dotyczą fizycznych możliwości plotera i nie mają wiele wspólnego z faktyczną prędkością kreślenia – zależy ona bezpośrednio od rodzaju pisaków (flamastry, rapidografy, pisaki kulkowe itp.) i papieru (kalka, papier, folia itp.) – patrz tabelka.

Słowniczek

CAD – Computer Aided Design
– projektowanie wspomagane komputerowo
drum plotter – ploter bębnowy
belt plotter – ploter pasowy
flatbed plotter – ploter stolikowy
rollerbed plotter – ploter rolkowy ("wyżymaczka")
axial speed – prędkość osiowa
diagonal speed – prędkość po przekątnej (wypadkowa)
acceleration – przyspieszenie
resolution – rozdzielczość
repeatibility – powtarzalność
accuracy – dokładność
dpi – dot per inch
– punktów na cal
ips – inch per second
– cali na sekundę
ADI – Autodesk Device Interface
– sprzętowy interface Autodesku
HPGL – Hewlett-Packard Graphic Language
– język graficzny Hewlett-Packarda
DM/PL – Digital Microprocesor/Plotting Language
– język graficzny Houston Instrument
ANSI – American National Standards Institut
– amerykański komitet normalizacji
ISO – International Standards Organisation
– międzynarodowy komitet normalizacji
DIN – Deutsche Industry Normen
– zachodnioniemiecki komitet normalizacji





Przyspieszenie

Przyspieszenie określa, jak szybko ruszający pisak osiąga maksymalną prędkość, gdy jest podniesiony lub opuszczony. Przyspieszenie podawane jest jako wielokrotność przyspieszenia ziemskiego (g) równego 9,81 cm/s2 i zawiera się zwykle w granicach 0,5 – 4g. Przyspieszenia ploterów z silnikami krokowymi nie przekraczają zwykle 1g. Efektywne przyspieszenie przy rysowaniu (opuszczony pisak) limitowane jest oczywiście możliwościami pisaka.

Opuszczanie/podnoszenie pisaka

Pisak opuszczany i dociskany jest zwykle do papieru za pomocą elektromagnesu, a podnoszony sprężyną powrotną. Droższe modele ploterów za pomocą amortyzatorów pneumatycznych lub elektromagnetycznych kontrolują siłę, z jaką pisak stawiany jest na papier – zwiększa to trwałość pisaków i zmniejsza hałas.

Efektywna prędkość kreślenia

Praktyczna prędkość kreślenia rysunków jest wypadkową wszystkich omówionych wyżej parametrów technicznych, formatu rysunku oraz szybkości zmian pisaków (w ploterach wielopisakowych), a także szybkości pracy komputera przygotowującego (wektoryzującego) rysunek.

Rozdzielczość

Rozdzielczość to najmniejsza odległość o jaką może przesunąć się pisak. Im większa rozdzielczość czyli mniejsza odległość, tym

> 44

43 <

gładsze są rysowane linie skośne i krzywe. Rozróżniana jest tzw. rozdzielczość sprzętowa czyli możliwość samego plotera oraz rozdzielczość programowa czyli zdolność programu do sterowania ploterem. W reklamach podawana jest rozdzielczość sprzętowa i wynosi ona zwykle 0,001 cala (0,025 mm), co odpowiada rozdzielczości drukarki równej 1000 dpi (punktów/cal) – drukarki laserowe mają np. rozdzielczość 300 dpi.

Rozdzielczość programowa jest zwykle dużo większa i nie stwarza komplikacji. Jej wpływ wystąpić może przy bardzo dużych (długich) rysunkach – ponad 2 m – i tak np. driver ADI do AutoCAD-a pozwala na max. 65535 kroków wzdłuż każdej osi.

Powtarzalność

Powtarzalność to maksymalny błąd, jaki wystąpi przy przesunięciu pisaka z jednego punktu do drugiego i z powrotem. Jest miarą wewnętrznych luzów mechanizmów plotera. Powtarzalność podawana jest w jednostkach długości i jej wartość nie powinna być większa od 0,004 cala (0,1 mm), im mniejsza tym oczywiście lepiej.

Na powtarzalność ma również wpływ jakość i precyzja wykonania pisaka – czy końcówka pisząca znajduje się dokładnie w środku (w osi) pisaka. Przy ploterach wielopisakowych powtarzalność zmniejsza się wraz ze zmianą pisaków, dodatkowo każdy z nich może mieć inaczej (mniej lub bardziej osiowo) umieszczoną końcówkę piszącą. Najgorszą sławą pod tym względem cieszą się pisaki kulkowe.

Dokładność

Jest to najważniejszy parametr charakteryzujący jakość plotera. Dokładność to maksymalny błąd narysowania punktów końcowych linii. Dokładność podawana jest jako błąd bezwzględny (w jednostkach długości) i jako błąd względny czyli procent długości rysowanej linii. Wartości poniżej 0,01 cala (0,254 mm) i 0,1% uznać należy za bardzo dobre.

Warto zauważyć, że ploter o dobrej powtarzalności może być niedokładny powielając ciągle ten sam błąd.

Formaty arkuszy rysunkowych		
Amerykański standard inżynierski (ANSI)		
nazwa	cale	milimetry (około)
A	8,5 x 11	216 x 279
B	11 x 17	279 x 432
C	17 x 22	432 x 558
D	22 x 34	558 x 864
E	34 x 44	864 x 1116
Europejski standard metryczny (ISO/DIN)		
A4	8,3 x 11.7	210 x 297
A3	11,7 x 18.5	297 x 420
A2	18,5 x 23.4	420 x 594
A1	23,4 x 33.1	594 x 840
A0	33,1 x 48.8	840 x 1188

Rodzaj pisaka	Zalecana szybkość kreślenia cal/s	Zalecana szybkość kreślenia mm/s	Uwagi
flamaster:			
– wodny	18	450	zamalowywanie konturów
– olejowy	4	100	rysowanie na folii
pisak kulkowy:			
– wodny	18	450	rysunki kontrolne,
– olejowy	18	450	
pisak ceramiczny	4	100	cienkie linie
rapidograf:			
– jednorazowy	8	200	stała grubość linii
– napełnialny	8	200	trzeba dokładnie czyścić

Język – sterowanie plotera

Każdy ploter ma wbudowany język graficzny. Umożliwia on łatwą kontrolę programu komputerowego nad pracą plotera. Język ten tłumaczy otrzymane rozkazy na sekwencje impulsów elektrycznych sterujących bezpośrednio pracą mechanizmów plotera. Obowiązują tu, jako standard, dwa języki liderów – HPGL (ang. *Hewlett-Packard Graphic Language*) firmy Hewlett- Packard i DM/PL (ang. *Digital Microprocesor/Plotting Language*) firmy Houston Instrument. Nie przyjął się natomiast jako standard protokół CalComp 906, równie znanego producenta ploterów. Tańsze plotery emulują DM/PL jako prostszy w implementacji, z drugiej strony jednak pakiety typu CAD preferują HPGL jako dający większe możliwości. Dlatego też plotery Houston-a z najnowszej serii DMP-60 emulują także standard rywala – HPGL.

Większość ploterów ma wbudowane różne kroje pisma, kreskowania, rodzaje punktów i linii, procedury wygładzające kreślone łuki i okręgi oraz optymalizujące cały proces kreślenia. Niewiele programów potrafi z nich jednak skorzystać. I tak, by obsłużyć jak najwięcej ploterów, AutoCAD przejmuje większość kontroli nad ploterem, nie wykorzystując prawie wcale jego wbudowanych możliwości. W trakcie przygotowywania rysunku AutoCAD tłumaczy wszystkie jego elementy na sekwencje dłuższych lub krótszych linii prostych. AutoCAD dokonuje także optymalizacji rysunku np. kolejno rysowane są wszystkie elementy rysunku w poszczególnych kolorach, dzięki czemu każdy pisak (kolor) zmieniany jest tylko jeden raz. Także komunikacja z ploterem odbywa się z pominięciem DOS-a, co po pierwsze znacznie przyspiesza pracę, a po drugie zwalnia użytkownika od ustawiania parametrów transmisji komendą MODE. Wprowadzony niedawno driver ADI (ang. *Autodesk Device Interface*) dla ploterów był początkowo ignorowany przez producentów sprzętu - istnieje silna dominacja HPGL i DM/PL. Z biegiem czasu okazało się jednak, że dzięki ADI można sporo zyskać i producenci sprzętu zaczęli go stosować.

Możliwości dodatkowe

Komputer wysyła dane do plotera szybciej niż mogą być narysowane. Standardowo plotery wyposażone są w pamięć do 20 KB RAM, wykorzystywaną jako bufor danych otrzymywanych z komputera, co nieznacznie tylko skraca czas jego blokowania. By proces ten jeszcze przyspieszyć, niektóre plotery oferują możliwość zamontowania dodatkowej pamięci RAM, zwykle w granicach 1-4 MB, co pozwala na zmagazynowanie danych o dużym nawet rysunku i uwolnienie komputera.

Przydatną opcją jest także automatyczne zakrywanie przed wysychaniem nie używanych chwilowo pisaków – ułatwia to znacznie pracę eliminując konieczność ich rozpisywania.

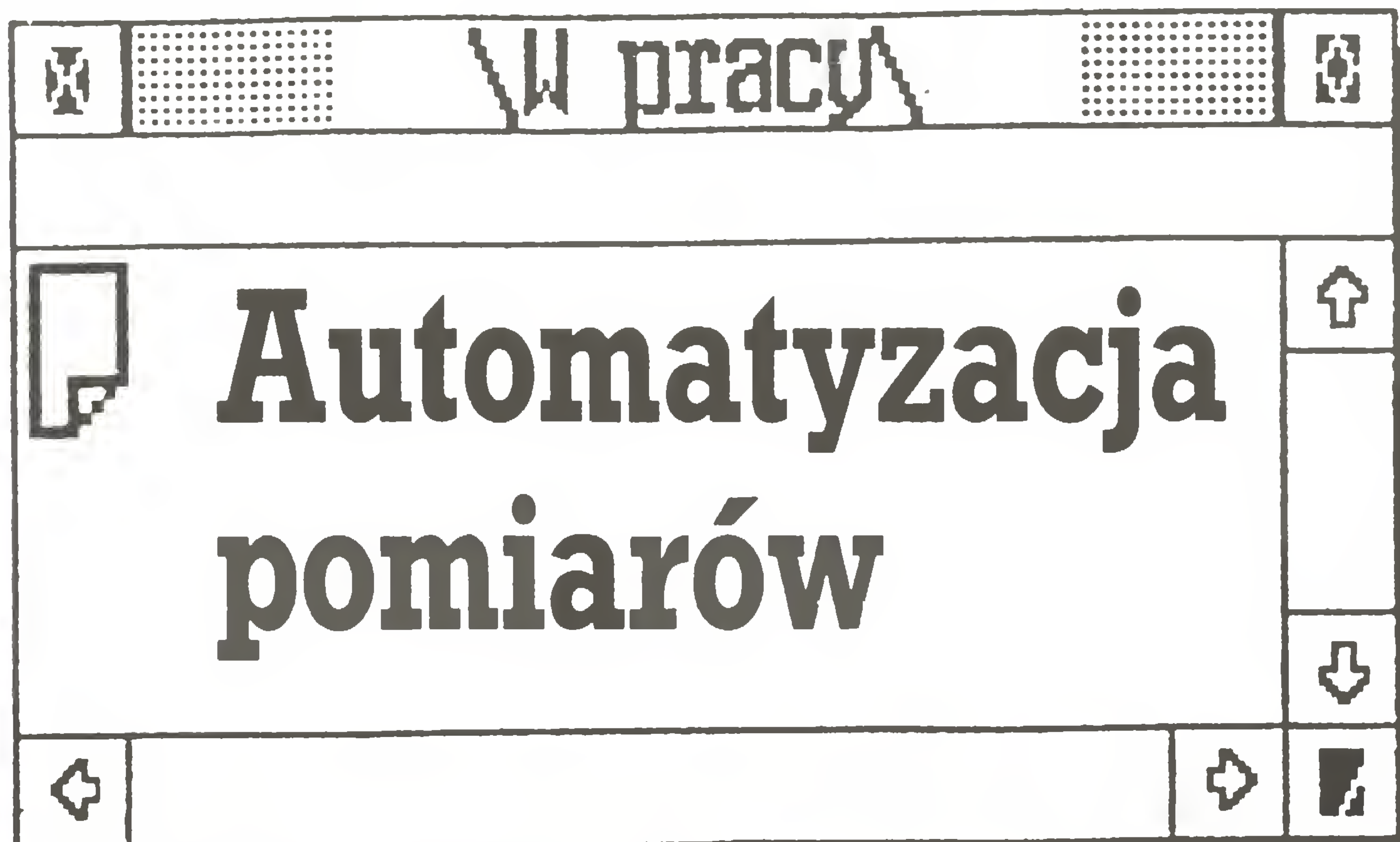
Plotery oferują zwykle także możliwość wykorzystania ich jako *digitizerów*. Opcja ta nie jest jednak zbyt wygodna w obsłudze i w codziennej pracy nie zastąpi na pewno prawdziwego *digitizera*.

Co kupić?

Oczywiście najlepsze na co nas stać. O cenie plotera, w największej mierze, decyduje jego format, precyzja wykonania i kreślenia oraz wytrzymałość. Jeśli rysujemy ponad 30 rysunków dużego formatu miesięcznie to kupić trzeba najwytrzymalszy (tzw. *heavy-duty*) i niestety najdroższy. Przy 20 rysunkach na miesiąc wystarczy trochę mniej wytrzymały i oczywiście tańszy. Przy jeszcze mniejszej liczbie rysunków kupić można coś jeszcze tańszego, choć trzeba pamiętać, że biednego nie stać na okazyjne kupno tanich bubli. Ważną sprawą jest także prostota obsługi, dostęp do materiałów eksploatacyjnych oraz serwis techniczny.

Mam nadzieję, że przedstawione w niniejszym tekście informacje pomogą w podejmowaniu decyzji o zakupie plotera.

W następnych numerach napiszemy jak testować plotery oraz zamieścimy test, udostępnionego nam przez spółkę APLIKOM, plotera Houston Instrument DMP-62.



Automatyzacja pomiarów wielkości fizycznych wiąże się ściśle z organizacją systemu pomiarowego i odpowiednim do niego wyborem interfejsu. On określa magistralę sterowania, transmisję danych i organizację przerwań. W Polsce najbardziej rozpowszechnione są standardy CAMAC i IEC-625 (IEEE 488). Wydaje się, że najmniejsze nakłady finansowe potrzebne są dzisiaj do wdrożenia systemu pomiarowego w standardzie IEC--625. Wszystkie elementy tego systemu można skompletować (za złotówki) w terminie nie dłuższym niż dwa miesiące. Oprócz swobody w wyborze sprzętu, użytkownicy mają możliwość korzystania z dużej ilości programów sterujących w językach: C, Pascal, Fortran i jeśli ktoś chce także w Basicu.

Standard IEC-625

Magistrala IEEE 488 jest przeznaczona do bezpośredniej współpracy aparatury pomiarowej ze sprzętem informatycznym. Jako standard została zdefiniowana przez IEEE (*Institute of Electrical and Electronic Engineers)* w 1975 roku z poprawkami przyjętymi w 1978 roku. Standard IEEE 488 jest inaczej nazwany: GPIB (ang. *General Purpose Interface Bus*), HP-IB (*Hewlett Packard Interface Bus*). W Polsce znany jest pod nazwą IEC-625 i określony normą PN-83/T-06536, w dalszej części będziemy używali tej nazwy.

Przed wprowadzeniem tego standardu wielu producentów aparatury oferowało swoje własne rozwiązania. To przenosiło ciężar projektowania interfejsów na użytkownika systemu pomiarowego. Jeżeli jego stanowisko pomiarowe zawierało wyroby różnych producentów, był on zmuszony do projektowania wielu interfejsów sprzętowych i programowych w celu połączenia tego w działającą całość. Swoją popularność interfejs IEC-625 (czasem nazywany GPIB) zawdzięcza swej pełnej specyfikacji elektrycznej i mechanicznej oraz protokołowi sterowania i transmisji danych. Zastosowanie standardu IEC- 625 zwolniło użytkownika od konieczności projektowania interfejsów i pozwoliło mu skoncentrować się na oprogramowaniu specyficznym dla konfiguracji sprzętowej zamierzonego eksperymentu.

Budowa

Głównym zadaniem magistrali IEC-625 jest przesyłanie informacji pomiędzy dwoma lub więcej urządzeniami. Przed rozpoczęciem transmisji konieczne jest określenie, które z urządzeń ma nadawać dane, a które odbierać. Decyduje o tym aktywny sterownik. Można go porównać z przewodniczącym zebrania. W magistrali IEC-625 tylko jedno urządzenie może nadawać w danej chwili. Nazywamy je wtedy aktywnym nadajnikiem. Fakt "udzielenia głosu" urządzeniu nazywamy zaadresowaniem na nadawanie (ang. Addressing to Talk).

Aktywny sterownik najpierw rozkazuje wszystkim urządzeniom nie słuchać, a następnie indywidualnie zwraca się do urządzeń, które mają przyjąć dane, by przełączyły się na odbiór. Nazywamy to zaadresowaniem na odbiór (ang. *Addressing to Listen*). W ten sposób dane trafią tylko do tych urządzeń, dla których są przeznaczone. Wszystkie pozostałe zignorują przesyłane dane.

W czasie gdy aktywny sterownik ustala nadajnik i odbiorniki, sygnał Attention jest aktywny, wskazując, że dane jeszcze nie są gotowe do transmisji. Aktywny sterownik zezwala na rozpoczęcie przesyłania danych zdejmując sygnał Attention z magistrali.

Podsumowując, aktywny sterownik sterując transmisją danych magistrali IEC-625:

a) generuje sygnał Attention, aby wstrzymać jakiekolwiek przesyłanie danych magistralą IEC-625;
b) rozadresowuje wszystkie urządzenia ze stanu odbioru, aby dane nie trafiły do niepożądanych odbiorników;
c) określa urządzenie, które ma nadawać, adresując je na nadawanie;
d) określa urządzenia, które mają przyjąć dane, adresując je na odbiór;
e) daje znać wszystkim urządzeniom, poprzez zdjęcie sygnału Attention z magistrali, że nastąpi transmisja danych.

Kiedy transmisja danych rozpoczęła się, aktywny nadajnik czeka, aż wszystkie odbiorniki będą gotowe na przyjęcie bajtu danych i dopiero wtedy wysyła pierwszy bajt do magistrali. Następnie sygnalizuje, że dane są gotowe do odbioru i czeka, aż wszystkie odbiorniki potwierdzą przyjęcie danych. Wtedy nadajnik wysyła kolejny bajt danych do magistrali i czeka, aż nadajniki znowu będą gotowe do przyjęcia danych. Kiedy to nastąpi, nadajnik informuje, że dane są gotowe do odbioru i cała sekwencja powtarza się do czasu zakończenia transmisji.

KODY ZNAKÓW ASCII I IEEE

wartość szesnastkowa → $56 | 86 ← wartość dziesiętna
znak ASCII → V
adres nadajnika IEEE → 22

ACG – Addressed Command Group
UCG – Universal Command Group
LAG – Listen Address Group
TAG – Talk Address Group
SCG – Secondary Command Group

ACG	UCG	LAG	LAG	TAG	TAG	SCG	SCG
$00 0 NUL	$10 16 DLE	$20 32 SP 00	$30 48 0 16	$40 64 @ 00	$50 80 P 16	$60 96 ` sco	$70 112 p sco
$01 1 SOH GTL	$11 17 DC1 LLO	$21 33 ! 01	$31 49 1 17	$41 65 A 01	$51 81 Q 17	$61 97 a sco	$71 113 q sco
$02 2 STX	$12 18 DC2	$22 34 " 02	$32 50 2 18	$42 66 B 02	$52 82 R 18	$62 98 b sco	$72 114 r sco
$03 3 ETX	$13 19 DC3	$23 35 # 03	$33 51 3 19	$43 67 C 03	$53 83 S 19	$63 99 c sco	$73 115 s sco
$04 4 EOT SDC	$14 20 DC4 DCL	$24 36 $ 04	$34 52 4 20	$44 68 D 04	$54 84 T 20	$64 100 d sco	$74 116 t sco
$05 5 ENQ PPC	$15 21 NAC PPU	$25 37 % 05	$35 53 5 21	$45 69 E 05	$55 85 U 21	$65 101 e sco	$75 117 u sco
$06 6 ACK	$16 22 SYN	$26 38 & 06	$36 54 6 22	$46 70 F 06	$56 86 V 22	$66 102 f sco	$76 118 v sco
$07 7 BEL	$17 23 ETB	$27 39 ' 07	$37 55 7 23	$47 71 G 07	$57 87 W 23	$67 103 g sco	$77 119 w sco
$08 8 BS GET	$18 24 CAN SPE	$28 40 (08	$38 56 8 24	$48 72 H 08	$58 88 X 24	$68 104 h sco	$78 120 x sco
$09 9 HT TCT	$19 25 EM SPD	$29 41) 09	$39 57 9 25	$49 73 I 09	$59 89 Y 25	$69 105 i sco	$79 121 y sco
$0A 10 LF	$1A 26 SUB	$2A 42 * 10	$3A 58 : 26	$4A 74 J 10	$5A 90 Z 26	$6A 106 j sco	$7A 122 z sco
$0B 11 VT	$1B 27 ESC	$2B 43 + 11	$3B 59 ; 27	$4B 75 K 11	$5B 91 [27	$6B 107 k sco	$7B 123 { sco
$0C 12 FF	$1C 28 FS	$2C 44 , 12	$3C 60 < 28	$4C 76 L 12	$5C 92 \ 28	$6C 108 l sco	$7C 124 ¦ sco
$0D 13 CR	$1D 29 GS	$2D 45 - 13	$3D 61 = 29	$4D 77 M 13	$5D 93] 29	$6D 109 m sco	$7D 125 } sco
$0E 14 SO	$1E 30 RS	$2E 46 . 14	$3E 62 > 30	$4E 78 N 14	$5E 94 ^ 30	$6E 110 n sco	$7E 126 ~ sco
$0F 15 SI	$1F 31 US	$2F 47 / 15	$3F 63 ? UNL	$4F 79 O 15	$5F 95 _ UNT	$6F 111 o sco	$7F 127 DEL sco

Linie magistrali

Magistrala IEC-625 składa się z 16 linii sygnałowych i 8 linii uziemienia. Osiem linii sygnałowych DI01 – DI08 (ang. *Data Input – Output*) tworzy magistralę danych, którą przesyłane są pomiędzy urządzeniami rozkazy i dane (bajt po bajcie). Trzy kolejne linie tworzą magistralę synchronizacji (ang. *handshake lines*), której zadaniem jest sterowanie sygnałami: "gotów na przyjęcie danych" (ang. *Ready For Data*), "dane ważne" (ang. *Data Valid*), "dane odebrane" (ang. *Data Accepted*) podczas transmisji danych. Pięć pozostałych to linie ogólnego zarządzania interfejsami (magistrala sterowania). Pełnią one szereg ważnych funkcji.

Informacja umieszczona w liniach danych może być bądź bajtem danych, bądź rozkazem. Jeżeli linia ogólnego zarządzania Attention (ATN) jest aktywna w czasie transmisji danych, wówczas na liniach danych znajduje się rozkaz wieloliniowy, który powinien być przyjęty przez każde urządzenie połączone z magistralą. W przeciwnym razie przesyłany jest

> 46

45 <

bajt danych i powinny go odebrać tylko aktywne odbiorniki.

Magistrala IEC-625 zawiera również pewną liczbę rozkazów jednoliniowych, które jak wskazuje ich nazwa, są przesyłane pojedynczymi liniami ogólnego sterowania. Na przykład linia Interface Clear (IFC), gdy jest aktywna, wysyła rozkaz Interface Clear do wszystkich urządzeń magistrali powodując ustawienie ich interfejsów w stan początkowy.

Adresowanie

Standard IEC-625 dopuszcza do 15 urządzeń w systemie. W celu uniknięcia konfliktów i nieporozumień każde urządzenie ma własny adres, który jest liczbą z przedziału od 0 do 30 i różni się od adresów innych urządzeń dołączonych do magistrali. Sposób, w jaki adres urządzenia jest ustawiany, określa producent. Niektóre adresy urządzeń ustawia się za pomocą przełączników typu DIP, inne w sposób programowy lub przełącznikami na płycie czołowej. Instrukcja obsługi urządzenia dostarczana przez producenta musi zawierać sposób ustawienia tego adresu.

Urządzenie zostaje zaadresowane na nadawanie, gdy przyjmie od aktywnego sterownika rozkaz wieloliniowy Talk Address Group (TAG) (bajt przesyłany przy aktywnym sygnale ATN) wraz ze swoim indywidualnym adresem. Podobnie zostaje ono zaadresowane na odbiór, gdy przyjmie rozkaz wieloliniowy Listen Address Group (LAG). Pozostałe rozkazy adresowe: My Talk Address (MTA), My Listen Address (MLA), rozkazy TAG i LAG dla aktywnego sterownika oraz Secondary Command Group (SCG) są wykorzystywane do odwołania się do adresów wewnętrznych oraz funkcji wewnętrznych w poszczególnym urządzeniu.

Sterownik systemu

Sterownik systemu jest urządzeniem, które zawsze ostatecznie odzyskuje sterowanie magistralą. Kiedy system jest po raz pierwszy włączony, sterownik systemu jest aktywnym sterownikiem i steruje wszystkimi operacjami magistrali. Sterownik systemu może przekazać sterowanie innemu urządzeniu czyniąc je aktywnym sterownikiem, on zaś z kolei może przekazać sterowanie kolejnemu urządzeniu. Nawet jeżeli sterownik systemu nie jest aktywnym sterownikiem, to obsługuje on linie ogólnego sterowania Interface Clear (IFC) oraz Remote Anable (REN) i dzięki temu może przejąć sterowanie magistralą, kiedy tylko zechce.

Linie ogólnego zarządzania magistralą

Pięć linii magistrali IEC-625 jest wykorzystywanych do jej sterowania. O sygnałach na tych liniach często mówimy jako o rozkazach jednoliniowych. Są one aktywne w stanie niskim, tzn. niski poziom napięcia odpowiada logicznej jedynce (stan ustawiony, załączony), zaś wysoki poziom napięcia logicznemu zeru. Niektóre z tych linii tworzą wspólny sygnał, tzn. są dołączone do jednej bramki typu OR. Niektóre linie mogą być ustawiane przez dowolne z urządzeń magistrali i sygnał na tych liniach będzie aktywny, gdy jakiekolwiek urządzenie go wysteruje. Na odwrót, sygnał nie będzie aktywny, jeżeli żadne z urządzeń go nie wysteruje. Magistralę sterowania tworzą następujące linie ogólnego sterowania interfejsów:

- Attention (ATN) (Uwaga)

ATN jest jedną z najważniejszych linii wykorzystywanych do sterowania magistralą. Kiedy ona jest aktywna (ATN=1, tj. linia w stanie niskim), informacje na liniach danych są interpretowane jako rozkaz, w przeciwnym razie traktuje się je jako dane dla Aktywnych Odbiorników. ATN jest sterowana przez aktywny sterownik.

- Interface Clear (IFC) (Zerowanie interfejsu)

Sterownik systemu wykorzystuje linię IFC do ustawienia wszystkich interfejsów urządzeń na magistrali w ustalony stan początkowy. Jednocześnie czyni sterownik Systemu sterownikiem Aktywnym.

- Remote Enable (REN) (Zdalne możliwe)

Sterownik systemu aktywizuje linię REN, aby umożliwić urządzeniom podłączonym do magistrali reakcję na rozkazy zdalnego sterowania. Jeżeli REN nie jest aktywna, urządzenia będą je ignorować (sterowanie ręczne).

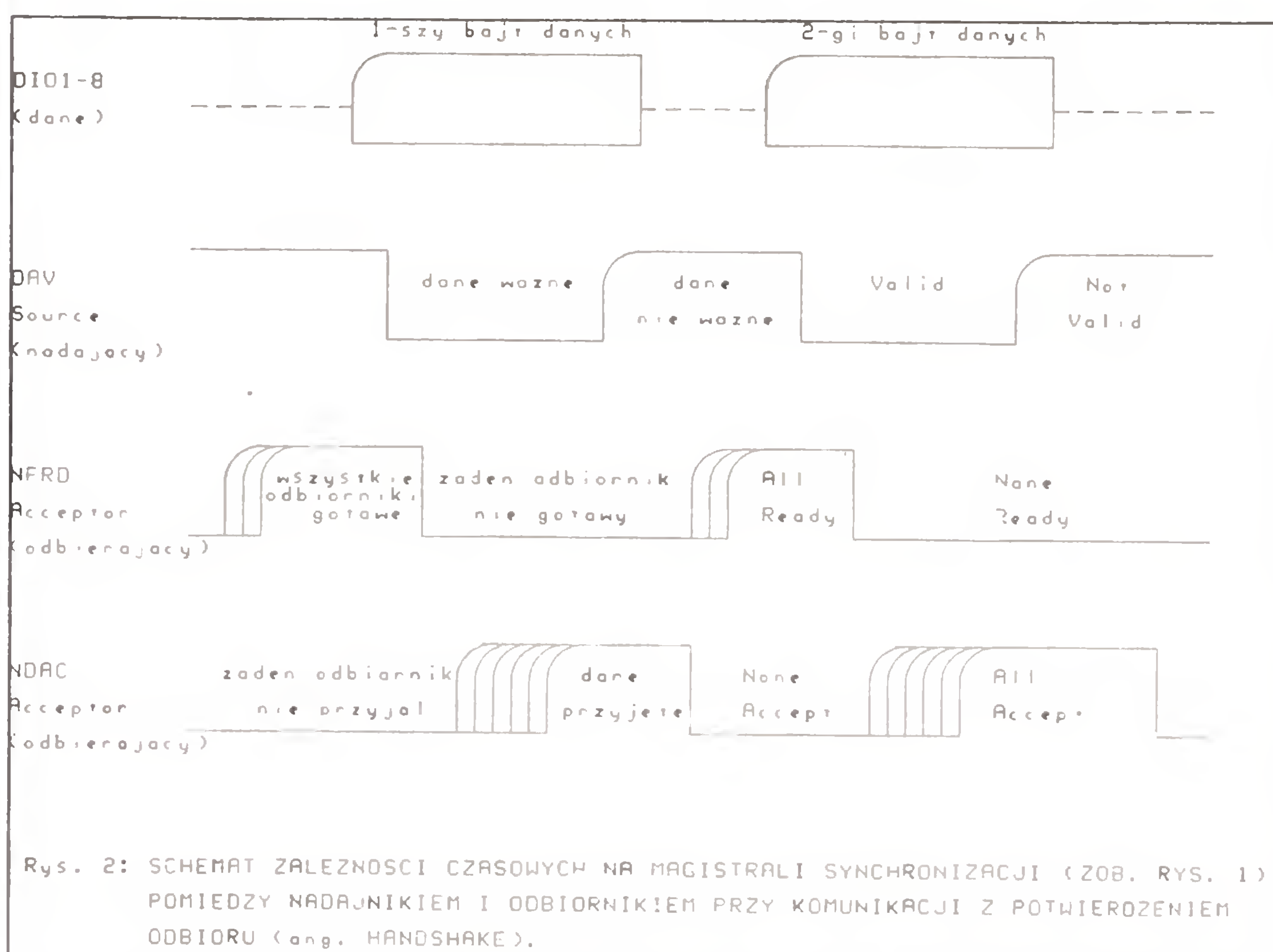
- End Or Identify (EOI) (Koniec lub identyfikacja)

EOI jest linią wykorzystywaną do zasygnalizowania ostatniego bajtu danych przy transmisji wielobajtowej. Urządzenie, które wysyła dane, uaktywnia EOI w czasie ostatniego bajtu. Korzystanie z sygnału EOI nie zawsze jest konieczne, gdyż ostatnim bajtem danych może być specjalny znak, taki jak CR czy LF.

Linia EOI jest wykorzystywana m.in. przez Aktywny sterownik do odpytywania równoległego (ang. *parallel poll*) poprzez jednoczesne ustawienie EOI i ATN w stan aktywny.

- Service Request (SRQ) (Żądanie obsługi)

SRQ jest wspólną linią ustawianą przez każde z urządzeń, które chce zwrócić na siebie uwagę Aktywnego sterownika. Sterownik może stwierdzić, które urządzenie żąda obsługi poprzez odpytywanie szeregowe lub równoległe. Sygnał SRQ może być wykorzystywany do przerwania pracy Sterownika i obsługi urządzenia, np. odczytywania wyniku.



Rys. 2: SCHEMAT ZALEZNOSCI CZASOWYCH NA MAGISTRALI SYNCHRONIZACJI (ZOB. RYS. 1) POMIEDZY NADAJNIKIEM I ODBIORNIKIEM PRZY KOMUNIKACJI Z POTWIERDZENIEM ODBIORU (ang. HANDSHAKE).

Linie przesyłania z potwierdzeniem odbioru

Magistrala IEC-625 wykorzystuje trzy linie do komunikacji z potwierdzeniem na zasadzie: "Jestem gotowy do przyjęcia danych – Nadaję dane – Odebrałem dane". Taki protokół komunikacyjny zapewnia poprawność przesyłania danych przy prędkości określonej przez najwolniejszy odbiornik. Nadajnik steruje jedną linią (DAV), zaś sygnały linii NRFD oraz NDAC są sumowane za pomocą bramek OR dla wszystkich aktywnych odbiorników. Linie te, podobnie jak inne linie IEC-625, są aktywne w stanie niskim.

- Data Valid (DAV) (Dane ważne)

DAV jest linią sterowaną przez aktywny nadajnik. Przed wysłaniem jakichkolwiek danych nadajnik sprawdza czy linia NDAC jest ustawiona (aktywna w stanie niskim), tzn. czy wszystkie odbiorniki przyjęły poprzedni bajt danych. Wtedy nadajnik wysyła bajt do magistrali danych i czeka do momentu, aż linia NRFD nie przejdzie w stan wysoki, tj. do chwili, aż wszystkie zaadresowane odbiorniki będą gotowe do przyjęcia informacji. Dopiero wtedy (NRFD=1 i NDAC=0) nadajnik ustawia linię DAV w stan aktywny (DAV=0), informując odbiorniki, że dane na magistrali są gotowe.

- Not Ready for Data (NRFD) (Nie gotowy na dane)

Linia NRFD jest wykorzystywana przez odbiorniki do zasygnalizowania nadajnikowi momentu, gdy będą gotowe do przyjęcia nowych danych. Nadajnik musi czekać na każdy z odbiorników, aż NRFD=1. To gwarantuje, że wszystkie urządzenia, które mają przyjąć informację, będą do tego przygotowane.

- Not Data Accepted (NDAC) (Dane nie przyjęte)

NDAC jest linią sterowaną przez odbiorniki, która informuje nadajnik, że wszystkie urządzenia zaadresowane na odbiór przyjęły informację. Każde urządzenie zwalnia sygnał NDAC (ustawia stan wysoki) z własną szybkością, lecz linia ta nie przejdzie w stan wysoki, dopóki najwolniejszy z odbiorników nie przyjmie bajtu danych.

Rozkazy wieloliniowe

Rozkazy wieloliniowe są przesyłane przez aktywny sterownik po magistrali danych przy aktywnej linii ATN. Można je podzielić na pięć grup:

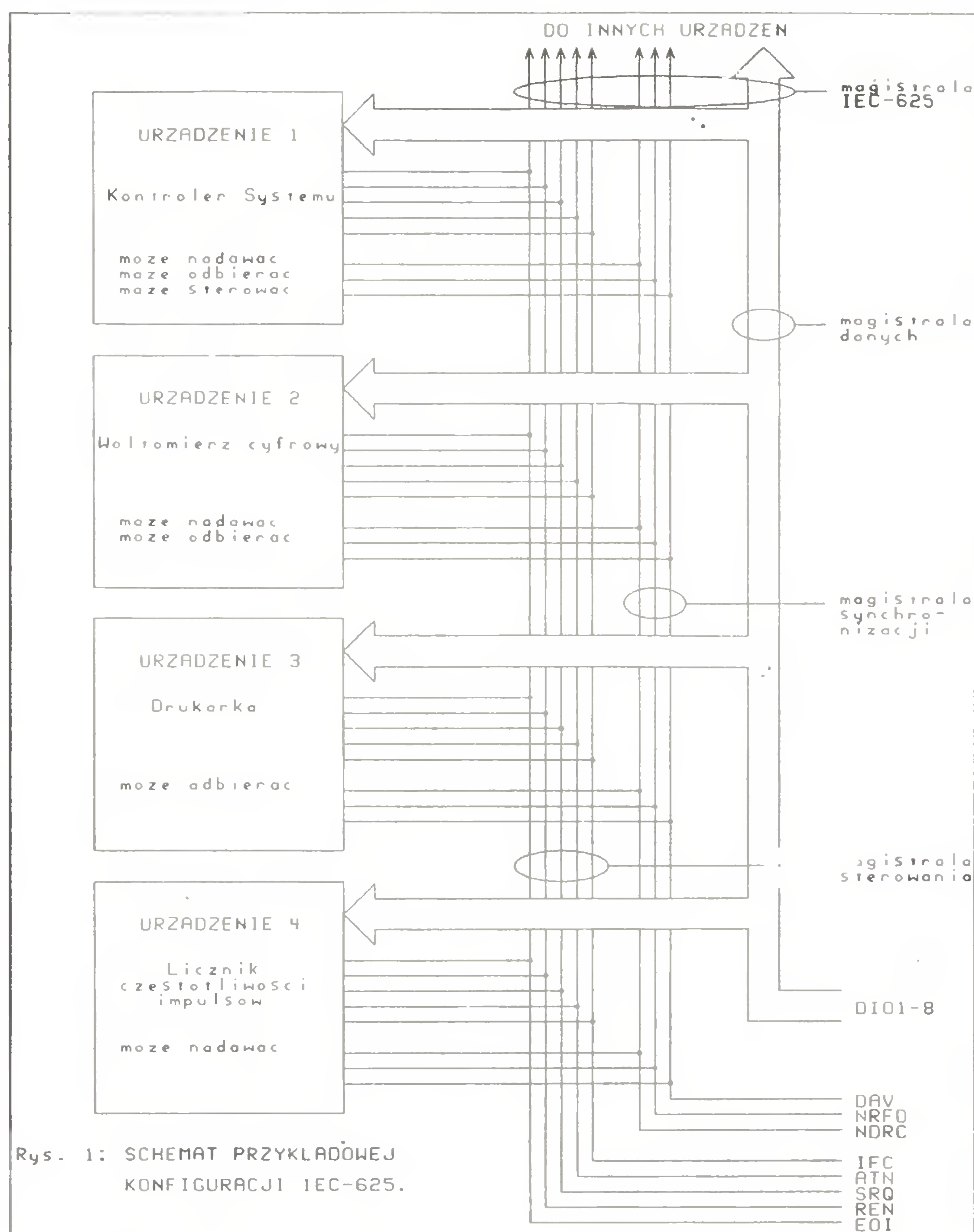
- grupę adresów odbioru (ang. *Listen Address Group* – LAG),
- grupę adresów nadawania (ang. *Talk Address Group* – TAG),
- grupę rozkazów wtórnych (ang. *Secondary Command Group* – SCG),
- grupę rozkazów adresowanych (ang. *Addressed Command Group* – ACG),
- grupę rozkazów uniwersalnych (ang. *Universal Command Group* – UCG).

- Listen Address Group (LAG) (20 – 3F) (zob. tabela)

Rozkazy ustawiają określone urządzenia w stan "odbiór" (ang. *Address to Listen*) (20 – 3E) lub wyprowadzają wszystkie urządzenia ze stanu odbioru (ang. *Unlisten* – UNL) (3F).

- Talk Address Group (TAG) (40 – 5F)

Rozkazy ustawiają określone urządzenia w stan "nadawanie" (ang. *Ad-*



Rys. 1: SCHEMAT PRZYKLADOWEJ KONFIGURACJI IEC-625.

dress to Talk) (40 – 5E) lub wyprowadzają wszystkie urządzenia ze stanu nadawania (ang. *Untalk* – UNT) (5F).

- Secondary Command Group (SCG) (60 – 7F)

Te rozkazy służą do określenia adresu wewnętrznego lub kodu funkcji wewnętrznej określonych urządzeń magistrali. Są one również wykorzystywane w sekwencji Parallel Poll Configure.

- Addressed Command Group (ACG) (00 – 0F)
- Universal Command Group (UCG) (10 – 1F)

Rozkazy te pełnią różnorodne funkcje. Rozkazy adresowane dotyczą tylko zaadresowanych odbiorników, podczas gdy rozkazy uniwersalne są przeznaczone dla wszystkich urządzeń magistrali.

- Go To Local (GTL) (ACG 01) (Przejdź na sterowanie lokalne)

Rozkaz GTL przywraca możliwość sterowania ręcznego wybranych urządzeń z płyt czołowych.

- Local Lockout (LLO) (UCG 11) (Blokada sterowania lokalnego)

Rozkaz LLO uniemożliwia jakiekolwiek ręczne sterowanie z płyty czołowej wszystkich urządzeń.

- Device Clear (DCL) (UCG 14) (Zerowanie urządzenia)
- Selected Device Clear (SDC) (ACG 04) (Zerowanie wybranego urządzenia)

Te rozkazy powodują ustawienie wszystkich (DCL) lub wybranych (SDC) urządzeń magistrali w zadany stan początkowy lub stan po załączeniu sieci.

- Serial Poll Enable (SPE) (UCG 18) (Odpytywanie szeregowe możliwe)

Rozkaz SPE umożliwia wszystkim urządzeniom wysyłanie bajtu stanu odpytywania szeregowego zamiast bajtu danych, w czasie gdy są Aktywnymi Nadajnikami.

- Serial Poll Disable (SPD) (UCG 19) (Odpytywanie szeregowe niemożliwe)

Rozkaz SPD zakazuje urządzeniom wysyłanie bajtu stanu odpytywania szeregowego.

- Group Execute Trigger (GET) (ACG 08) (Wyzwolenie grupy urządzeń)

Ten rozkaz zwykle wyzwala działanie pewnej grupy urządzeń. Umożliwia on jednoczesne rozpoczęcie pracy różnych urządzeń.

- Take Control (TCT) (ACG 09) (Przejmij sterowanie)

Rozkaz TCT przekazuje sterowanie magistralą innemu urządzeniu, które staje się wtedy aktywnym sterownikiem.

- Parallel Poll Configure (PPC) (ACG 05) (Zaadresowanie do odpytywania równoległego)

Rozkaz PPC przyporządkowuje urządzeniom numer bitu danych, który muszą wysłać w odpowiedzi na odpytywanie równoległe, gdy wymagają obsługi.

- Parallel Poll Unconfigure (PPU) (UCG 15) (Likwidacja odpytywania równoległego)

Ten rozkaz zabrania urządzeniom odpowiadać na odpytywanie równoległe.

Żądania obsługi

Linia SRQ jest ustawiana przez urządzenia, które wymagają uwagi aktywnego sterownika. W ten sposób urządzenie może dać znać sterownikowi, że zaszło jakieś zdarzenie. Przypomina to w pewien sposób wspólną linię przerwań w systemie mikroprocesorowym.

Większość inteligentnych urządzeń peryferyjnych ma możliwość ustawiania linii SRQ. Na przykład woltomierz cyfrowy mógłby ją ustawić, kiedy zakończy pomiar lub gdy nastąpiło przekroczenie zakresu. Zasilacz mógłby ustawić SRQ, gdy jego wyjście ma ograniczony prąd. Fakt możliwości zwolnienia aktywnego sterownika z konieczności okresowego sprawdzania stanu urządzeń na magistrali stanowi dużą zaletę. W zamian za to sterownik zleca urządzeniom: "Róbcie co wam nakazałem i dajcie mi znać, gdy skończycie lub gdy coś jest nie w porządku".

Ponieważ SRQ jest rozkazem jednoliniowym, więc sterownik nie ma możliwości określenia, które urządzenie żąda obsługi. Aby zidentyfikować źródło żądania obsługi, sterownik musi dokonać odpytywania szeregowego lub równoległego.

Odpytywanie szeregowe (ang. serial poll)

Kiedy urządzenie jest odpytywane szeregowo, zwraca sterownikowi bajt stanu, w którym najstarszy bit (DI07) będzie aktywny, jeżeli to urządzenie żądało obsługi, zaś pozostałe bity stanowią specyficzną informację o stanie. Jeżeli to urządzenie nie żąda obsługi, wówczas sterownik może odpytać szeregowo kolejne urządzenie, aż znajdzie to, które wymaga uwagi. W ten sposób powtarzane odpytywanie szeregowe pozwala ustalić, które urządzenie wymaga obsługi i jaki jest jego stan.

Odpytywanie równoległe (ang. parallel poll)

Kiedy sterownik dokonuje odpytywania równoległego, każde urządzenie wymagające obsługi ustawia jedną linię DI0 w stan aktywny. Badając stan tych linii sterownik może określić, które urządzenie żąda obsługi. Ponieważ nie otrzymuje on żadnej innej informacji o statusie urządzenia, więc często po odpytywaniu równoległym następuje odpytywanie szeregowe.

Odpytywanie szeregowe jest bardziej popularne od odpytywania równoległego, ze względu na możliwość odczytania stanu urządzenia. Ponadto większość urządzeń ma w zestawie funkcji tylko odpytywanie szeregowe.

Jacek Nogala
Jolanta Peryt
Marek Peryt
Lech Proboszcz
Mirosław Świniarski

Bibliografia:
1. "HP-IB Improving Measurements in Engineering and Manufacturing – a collection of useful technical information"
2. "IEEE Standard 488 – 1978. Digital interface for Programable Instrumentation". The IEEE, New York, N. Y. USA, Nov. 1978.
3. Polska Norma PN-83/T-06536 "System interfejsu dla programowanej aparatury pomiarowej. Przesył informacji bajty - szeregowo, bity – równolegle".
4. "Systemy interfejsu w miernictwie" praca zbiorowa pod kierunkiem Wojciecha Nowakowskiego, WKŁ, Warszawa 1987.
5. "IEEE 488 Technical Review" IOtech 1988 Catalog, First Edition.

WAKE U

Czas podjąć decyzję



LC 24-10

Drukarki Star możecie Państwo kupić osobiście w Hamburgu lub Berlinie Zachodnim:

ABC Computersystems GmbH
Alt Moabit 80
1000 Berlin 21
Berlin West
tel. 030/39 15 090/99
tlx. 18 13 65
fax. 030/39 36 483

ABC Data GmbH
Ditmar-Koel-Str. 13
2000 Hamburg 11
RFN
tel. 040/31 40 03
tlx. 21 66 002
fax. 040/31 91 783

LC-10	450.-DM	LC 24-10	770.-DM
LC-10 colour	550.-DM	XB 24-15	1350.-DM
LC-15,	750.-DM	XB 24-10	1150.-DM
FR-15	1200.-DM	LS-08(laserowa)	3900.-DM

star

Twoja drukarka



LC-10



NB 24-15



LS-08

Za transport do domu odbiorcy w Polsce prosimy doliczyć 40.- DM (za LS-08 100.-DM).

Wyłączny przedstawiciel na Polskę:
ABC Data GmbH
Augustastrasse 40
5300 Bonn 2, RFN
Tel. 0228/35-44-80,-90
Tx. 885566 abc bs

Ko-62/344/07

Obniżka cen Drukarka **STAR LC 10** już tylko **350 DM** (ok. 175 USD). Gdzie? **W ABC Data** Bonn, Berlin, Hamburg. Oczywiście z roczną gwarancją.



ZAKŁADY ELEKTRONICZNE

"MICRONET"

81-836 Sopot, ul. Krasickiego 9
tel. 51-13-17, tlx 051-2876

oferują

TERMINAL MT-220

funkcjonalnie zgodny z terminalem VT220 firmy DEC

MT-220 – emuluje terminale VT52, VT100, VT200 oraz PC-Shadow
MT-220 – posiada możliwość współpracy z dowolną drukarką wyposażoną w złącze równoległe lub szeregowe
MT-220 – może być stosowany w zestawach mikrokomputerowych (np: IBM PC/XT, IBM PC/AT), jak również jako końcówka do większych maszyn (np: SM-3, SM-4, SM-5, PDP-11, MERA 400)
- tryb VT52, VT100, VT200 polecany do pracy pod kontrolą systemów operacyjnych XENIX, UNIX, QNX, RSX, RT-11,
- tryb PC-Shadow zalecany do pracy pod kontrolą systemu typu MultiLink, PC-MOS

MT-220 – umożliwia wybór emulowanego terminala oraz parametrów jego pracy w prosty sposób przez samego użytkownika
MT-220 – sprzedawany jest w zestawie: monitor monochromatyczny z poświatą bursztynową i klawiaturą typu IBM PC/AT produkcji zachodniej
MT-220 – wyróżnia spośród innych konstrukcji tego typu spotykanych w Polsce wysoka jakość obrazu oraz niska cena.
MT-220 – to konstrukcja oparta na własnych rozwiązaniach technicznych i sprawdzona w dwuletniej eksploatacji

ZAPRASZAMY

Ko-193/302/04

Videcom® Sp. z o.o.
tel. 214662

chcesz kupić
IBM PC XT/AT,
twardy dysk 120MB?
nie śpiesz się!
lepiej wypożycz!

Warszawa, ul. Marszałkowska 72/10

Procedury dostępu
do plików dBaseIII i Clipper
z poziomu Pascala Turbo

oferuje

– U.I. R.Brykajło,
Kraków tel. 55-31-00 wew. 10-22.

Ko-118/260/03

NAPRAWA ZASILACZY
do IBM XT/AT

SP. RZEM. "CENTRUM"

Warszawa
tel. 610 71 80, 49-28-12

Ko-157/290/04

TURBO PASCAL 5.0
(750 str.) po polsku,
oraz **TURBO C 2.0,**
TURBO ASSEMBLER 1.0 i inne
poleca:
SYNTAX,
Kraków, ul. Langiewicza 10/3,
tel. 43 39 57.

Ko-212/317/05

REGENERACJA KASET
DO DRUKAREK
TUSZ USA
RACHUNKI
23-11-16 **"REFIL"** 8-11
Korespondencja:
ul. 20-Latków 1/79
02-157 Warszawa

Ko-242/335/06

Zadbaj o przyszłość swojej firmy.

Zautomatyzowany System Zarządzania

– sieciowe i wielodostępne środowisko (dostęp do danych na poziomie rekordu),
– równoczesna i równorzędna praca wszystkich stanowisk,
– wspólna baza danych,
– modularne podsystemy:

- Księga Główna,
- Gospodarka Materiałowa,
- Gospodarka Finansowa,
- Kadry,
- Informowanie Kierownictwa,
- Płace,
- Środki Trwałe,

– 2 nagrody na Targach Softarg '88.



BIURO PROJEKTÓW I ZASTOSOWAŃ SYSTEMÓW INFORMATYCZNYCH
MicroSystem Sp. z o.o. od 1985 j.g.u.

80-557 Gdańsk
ul. Marynarki Polskiej 59
tel. 43-05-24, 43-12-71 do 81 wew. 450
tlx. 0512749 hwspl

Ko-208/327/05



INTERSOFT

Firma oferuje sprzęt komputerowy klasy IBM PC, AMSTRAD.

Komputerowe systemy obsługi magazynów sklepowych, kasy sklepowe, metkownice

Dokumentację w języku polskim do komputerów IBM:

Uwaga Wirus (jak ustrzec się wirusów)	1.500 zł
Przewodnik programisty IBM	80.000 zł
Wprowadzenie do komputerów IBM	23.000 zł
Framework IIp	94.000 zł
System operacyjny DOS 3.30	85.000 zł
System operacyjny DOS 4.0	110.000 zł
GW Basic kompilator	38.000 zł
Programowanie w GW Basic	90.000 zł
Turbo Basic 1.0	105.000 zł
Turbo C (200 atrakcyjnych procedur)	50.000 zł
Aztec C v. 3.2	99.000 zł
Język C dla zaawansowanych	110.000 zł
Turbo Graphic do TP 3.0	30.000 zł
Turbo Debugger, Turbo Assembler	63.000 zł
Turbo Pascal wersja 5.0	120.000 zł
Turbo Database Toolbox	25.000 zł
Grafika Turbo Pascala 4.0 i Turbo C 1.5	69.000 zł
Metody numeryczne do Turbo Pascala 4.0/5.0	75.000 zł
Turbo Power Tools (procedury do Turbo Pascala 4/5)	85.000 zł
Procedury numeryczne do Fortranu SSP/PC	80.000 zł
Przewodnik zaawansowanego programisty do dBase II/III	80.000 zł
dBase III poradnik encyklopedyczny	110.000 zł
dBase III + programowanie	60.000 zł
dBase III + zastosowania	90.000 zł
dBase III + poznawanie	55.000 zł
dBase III + opis pakietu sieciowego	30.000 zł
dBase III + instalacja	13.000 zł
dBase III + generator aplikacji	10.000 zł
dBase IV (Co nowego w dBase IV)	75.000 zł
Clipper 87, kompilator do dBase III	120.000 zł
Praktyka programowania w Clipperze	90.000 zł
Fox Base	85.000 zł
Eureka	56.000 zł
PC Tools De Lux	20.000 zł
Stat Graphics	70.000 zł
Programowanie w Assemblerze	110.000 zł
Instrukcja obsługi PC 1512	65.000 zł
Chi II Writer / WordStar	15.000 zł
Wordstar 2000	38.000 zł
Instrukcja do drukarki NX 15	40.000 zł
Norton Commander 87	40.000 zł
Or Cad	110.000 zł
Opis systemu OS 2	90.000 zł
Novell – podręcznik użytkownika	120.000 zł
Novell – podręcznik instalatora	90.000 zł
Instalacja karty 3 Com Etherlink	32.000 zł
Instalacja karty Arcnet	32.000 zł
POLONUS, spolszczona wersja dBase III+, kompunikaty, sortowanie i indeksowanie po polsku (kody Mazovii), CGA lub Hercules, 3 dysk., 322str. instr	289.000 zł
CLIPPER SARMATA 87, spolszczona wersja kompilatora do POLONUSA, 463 str., instr. 3 dyski	339.000 zł
FRAMEWORK IIp, pakiet zintegrowany, polskie litery (kody Mazovii) wymienność programów z POLONUSem, CGA, EGA lub Hercules, 4 dysk., 314 str., ins r.	179 000 zł
DRUKARZ, program pomocniczy do generacji wydruków, także specjalna wersja do programu FRAMEWORK IIp, 1 dyskietka, 27 str., instr	59.000 zł
ELEKDRUK, spolszczona wersja programu Smartwork, projektowanie prostych płyterk drukowanych, praca tylko na CGA, 1 dyskietka, 41 str., instr.	59.000 zł
WYKRES, spolszczona wersja programu MS Chart, możliwość wykonywania profesjonalnych wykresów i zestawień, CGA lub Hercules, 2 dyskietki, 411 str., instr.	150.000 zł
PISMAK, polski edytor tekstu, znacznie szybszy od Chi Writera	250.000 zł
KARTA, rozliczenie bazy transportowej	1.500.000 zł
PAW – program profilaktyki antywirusowej, wykrywa wszystkie znane do tej pory wirusy komputerowe	300.000 zł
PROGRAMOWA INSTALACJA ZNAKÓW – instalacja zaprojektowanych przez użytkownika znaków na ekranie i drukarce, współpraca z dowolnym oprogramowaniem	250.000 zł
CLIPPER NA UŻYTEK POLSKI nakładka na Clippera (biblioteka), pozwalająca sortować i indeksować bazy danych zgodnie z polskim alfabetem. Współpracuje z powyższym programem	240.000 zł
dB_DG – program do generacji dużych testowych baz danych, pomaga przy tworzeniu oprogramowania	200.000 zł

Zniżka 10%

przy płatności czekiem lub gotówką

Firma posiada uprawnienia do prowadzenia handlu zagranicznego.

Adres:
00-443 Warszawa,
ul. Górnośląska 9/11
telefon 21-56-08, 28-67-94, telex 817245

Dla akwizytorów atrakcyjne prowizje!

Ko 58/300/07



Zakłady Produkcyjno-Usługowe "WOLA" Sp. z o.o.

(jednostka gospodarki uspołecznionej),

00-726 Warszawa 36, box 40, ul. Willowa 8/10
tel: 48-03-05, 49-56-66, tlx 816264

Oddział w Toruniu:
87-100 Toruń, ul. Rydygiera 1d m.4, tel. 48-01-44

Oferują do sprzedaży:

Mikrokomputery IBM:
PC/XT/AT/386 firmy Future Systems w dowolnej konfiguracji

Urządzenia peryferyjne
Sprzęt wideo
Telefaxy

Realizujemy zamówienia na wszelki specjalistyczny sprzęt elektroniczny i duże partie podzespołów. Udzielamy gwarancji, zapewniamy serwis pogwarancyjny i materiały eksploatacyjne.

Ko-65/225/01

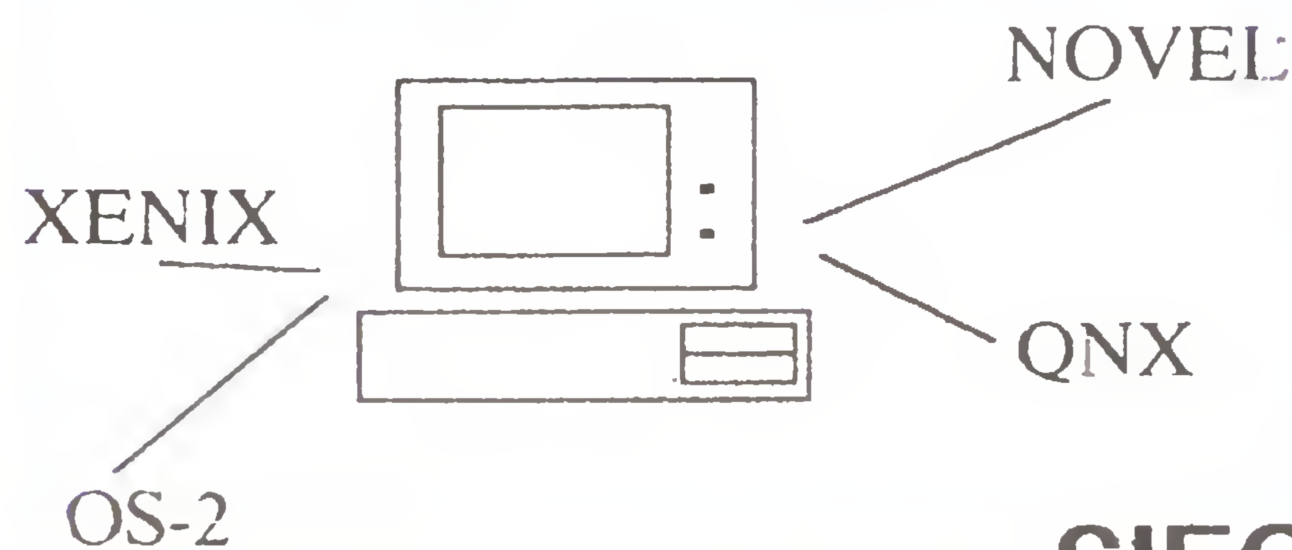


TEL. 19-43-91
TLX 815917 ZEGWA

KOMPUTERY	XT • AT • 386
DRUKARKI	STAR
PLOTERY	ROLAND
DIGITIZERY	SUMMAGRAPHICS
ELEMENTY	☆ ☆ ☆

COMERS ELECTRONIC

WIELODOSTEP

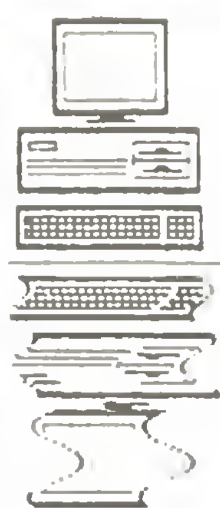


SIECI



UL. ZAMOYSKIEGO 2
03-801 WARSZAWA






Logic

Ogłoszenia przyjmuje Biuro Ogłoszeń, ul. Poznańska 38, 00–689 Warszawa, tel. 29 83 28


atomica
TURBO RESET
12
MHz
atomica
Ultra Slim
ABC
ABC
EGA MODE